Probabilistic Modelling of Station Locations in Bicycle-Sharing Systems

Daniël Reijsbergen

University of Edinburgh
`dreijsbe@inf.ed.ac.uk`

Abstract. We present a simulation methodology for generating the locations of stations in Bicycle-Sharing Systems. We present several methods that are inspired by the literature on spatial point processes. We evaluate how the artificially generated systems compare to existing systems through a case study involving 11 cities worldwide. The method that is found to perform best is a data-driven approach in which we use a dataset of places of interest in the city to 'rate' how attractive city areas are for station placement. The presented methods use only non-proprietary data readily available via the Internet.

1 Introduction

Bicycle-Sharing Systems[1] (BSSs) are an increasingly popular phenomenon, as witnessed by the worldwide number of operational systems growing from roughly 350 such systems [7] in 2010 to almost 1,000 at the time of writing [18]. Due to this strong increase, the question of choosing station locations in a new BSS is of increasing relevance to planners, operators, and scientists. A well-studied approach is to optimise some measure of coverage of 'interesting' parts of the city. The choice of Geographic Information System (GIS) dataset to identify interesting locations in the target city depends on the context — e.g., residential and commercial area density are of interest to a commuting-oriented BSS, and proximity to landmarks and restaurants to a BSS focussing on tourism and leisure. In the literature, the best station locations are then typically chosen using some form of deterministic optimisation — e.g., the methods implemented in the geographic analysis tool ArcGIS [10] or the optimisation tool suite XPRESS [8]. This approach will typically return a single optimal solution. In some cases, the user is required to manually determine a set of candidate locations beforehand.

In this paper, we will consider the use of stochastic simulation to generate BSS station locations as an alternative to deterministic optimisation. The probabilities informing the simulation procedure are inspired by the literature on spatial point processes. We will discuss the use of two baseline approaches — in particular the Poisson and Ginibre point process — and compare these to a simulation procedure that incorporates GIS information. The main criterion is correspondence to real-world BSSs, including several major systems such as the

---

[1] Alternatively called Bicycle-Sharing Plans.

Bicing BSS in Barcelona. The simulation procedure provides insight into how stations are distributed across space - this can inform models of bicycle movement, provide feedback to operators of existing systems, and aid designers of new systems. Executing this approach several times will result in different outcomes, which is typically more informative to planners than the single solution returned by a deterministic optimisation procedure. After all, the GIS data informing the optimality criterion is itself prone to subjectivity and inaccuracies. Additionally, the approach discussed in this paper only uses non-proprietary data and the programming code for the experiments is written in Java and available upon request.

The outline of this paper is as follows. We begin with a discussion of related work and the data sources used in this paper in Section 2. We then fix notation and discuss basic simulation procedures in Section 3. In Section 4, we zoom in on the various techniques to generate station locations. We present the results of a simulation experiment involving a comparison with real-world BSSs in Section 5. Section 6 concludes the paper.

2 Background & Data

Before we present this paper's specific contributions, we first elaborate on its position within the wider scientific and societal context. We begin with a discussion of the background and the related scientific literature in Section 2.1, and discuss the data sources used to generate the results of this paper in Section 2.2.

2.1 Background & Related Work

The recent wave of attention for BSSs from the scientific community is largely due to vast amount of data collected by *third-generation* systems [7]. Third-generation systems combat the theft and vandalism that plagued the previous two generations by employing technologies that allow for bikes and users to be uniquely identified. Data regarding bike availability at stations is collected as a by-product of such systems, and in many cases made available to third-party users, including researchers. One popular research area involves the analysis of bicycle usage patterns [9, 11, 13, 20]. Another area, one that is particularly relevant to his paper, involves the development of algorithms for the positioning of station locations [10,24]. These algorithms can be validated against real-world systems using the station location coordinates that are often provided alongside bike availability data. In the following we present a brief overview of related work concerning both the evaluation of existing systems in terms of their station locations and the design of new systems.

A comprehensive comparison of 38 BSSs worldwide is presented in [21]. Although some of the metrics considered in [21] correspond to usage patterns, some of the metrics solely deal with the geographic position of the stations - two of these metrics will also be considered in Section 5. Another research question involves the clustering of stations with similar usage patterns — case studies

include Barcelona [9], London [16], and Paris [4]. Of particular interest to this paper is the question of where to position the stations, given a GIS dataset to identify interesting locations: [10] does this for a proposed system in Madrid. The used GIS datasets include the road network, building usages in terms of population and jobs, 'transport zones', and locations of other public transportation hotspots. In [3], a methodology for adding new stations to BSSs is discussed and applied to two BSSs (Washington DC and Hangzhou) — the used GIS datasets here include the Google Places API for identifying Points of Interests (POIs), Location Based Social Network check-in obtained via the Foursquare API, and a demographical dataset. The potential introduction of a BSS in Helsinki, Finland is studied in [15], with a particular focus on open data. A potential BSS in Coimbra, Portugal is studied in [8]. The BSS in New York is studied in [17]; one of the GIS datasets used to identify locations of interest corresponds to taxi GPS transactions. A case study involving data provided by the municipality of Milan is presented in [5]. Naturally, the techniques used to optimise station locations can be applied in many other contexts a well — e.g., placement of defibrillators [22], fire watchtowers [2] or electric taxi charging stations [23] — see also the overview presented in the section on location-allocation models in [10]. For a general overview of location-allocation modelling of bike sharing systems can be found in [24]. Finally, in [12], a methodology for automatically learning patterns in a spatial context is discussed.

2.2   Data

Two main data sources are used in this paper. We use the OpenStreetMap project [14] as a source of geographical information, and use the CityBikes API [1] to obtain data about the locations of bicycle-sharing stations around the world. Both are discussed in more detail below.

OpenStreetMap OpenStreetMap (OSM) is a project in which users collaborate to create a dataset of roads and places that forms the basis of a nonproprietary map of the world. The database consists of four types of entries: *nodes*, *ways*, *relations* and *tags*. The *nodes* are single latitude-longitude coordinates indicating, e.g., points of interest or corners of larger areas. *Ways* are sequences of nodes, denoting paths and polygons that determine, e.g., the shapes of parks and roads. *Relations* are ordered lists of nodes, ways and other relations, and are used to denote larger geographic entities such as cycling routes or large motorways. The *tags* are used to store metadata about the other three data types. A tag can be used to, for example, indicate that a node corresponds to a convenience store or highway traffic lights.

   We use the OSM database for two purposes. First of all, we use it to identify places that are unsuited for BSS station placement, as we discuss in Section 3.1. A second purpose for the OSM database is that we use it to collect data about places of interest. In particular, we are interested in nodes that are tagged as a 'shop' or 'amenity'. Note that the latter category is relatively broad, and includes

park benches among others. Still we use this as a way of identifying locations of interest, as we discuss in greater detail in Section 4.4. The idea behind this choice is that those locations indicate areas that are interesting from a leisure-oriented point of view.

CityBikes To obtain data about the station locations of existing BSSs, we use the API for the website `citybik.es`. The project behind this website started as an Android app named CityBikes which helped users plan journeys in a BSS by displaying information about station occupancies. The data used by the app has been made available through a publicly-accessible API. Their system features information about BSSs worldwide, although we will only consider a fraction of those in this paper.

3 Preliminaries

This section combines several topics underlying all of the BSS station placement algorithms of Section 4. The structure of this section is as follows. In Section 3.1, we discuss the choice of a target area given that a city has been selected. We discuss ways to denote and characterise a configuration of BSS station locations — from now on referred to as a *topology* — in this city in Section 3.2, and then discuss a generic simulation procedure for generating a BSS in Section 3.3.

3.1 Target Areas and Valid Station Placement Locations

Given a choice of city, the first step in obtaining a BSS topology is determining the *target area*, i.e., the area within the city in which BSS stations can be placed. Initially, this will be (roughly, considering the approximately ellipsoidal shape of the face of the earth) a rectangle $[\lambda_{\min}, \lambda_{\max}, \phi_{\min}, \phi_{\max}]$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum longitudes respectively and $\phi_{\min}$ and $\phi_{\max}$ the minimum and maximum latitudes. The choice of target area has an impact of the accuracy of the method — as we discuss below, we discretise space by projecting the (approximate) rectangles spanning the city onto pixels in an image file. Since we are limited by memory constraints, there is a trade-off between the zoom level (i.e., how big the rectangles are — smaller rectangles capture more detail) and the size of the target area. Since we compare simulation procedures to existing BSSs in this paper, we choose the initial target areas to match the bounding boxes of the existing BSS's station locations, with a margin of 1km added. If one were interested in designing a BSS for a city that does not currently have one, manual selection of a rectangle as a target area, possibly informed by opening OSM in a web browser, would be the most straightforward approach.

    Within a typical initial target area, not all areas will be suited for station placement — particularly bodies of water, parks/forests, and farmland/desert. To identify those areas, we download the pre-generated 256-by-256-pixel tiles used to display OSM maps in a browser. The tiles in the full OSM dataset span the whole world and are generated for each of 20 zoom levels, where level 0
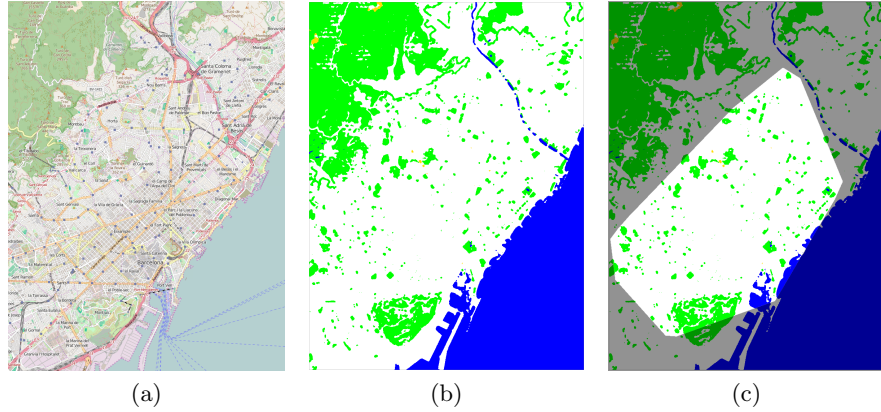
Fig. 1. Original and filtered versions of the OSM map of the target area in Barcelona. The figure on the far right includes the convex hull spanned by 100m circles around the stations in the Bicing BSS.

corresponds to capturing the entire face of the planet in a single 256-by-256-pixel tile and each subsequent level zooms in further by a factor 2. Given an initial target area in the form of a rectangle, we first identify the tiles corresponding to this area, and obtain an appropriate zoom level. This is done by specifying a maximum of number of pixels in the final image, and then determining the highest zoom level such that the resulting map still has fewer pixels than the maximum. When this is done, we amalgamate all the needed tiles in a single PNG image file — see Figure 1a for an example involving the Bicing BSS in Barcelona. Finally, we identify the pixels corresponding to bodies of water in the following way: we identify pixel colours ranges[2] that tend to correspond to water bodies, then we identify for each pixel whether it falls inside that range, and finally mark a pixel as 'water' if there are 8 of those pixels within a circle with a radius of 3 pixels around it. These pixels are marked as blue in the filtered image depicted in Figure 1b. We apply a similar pictures with green pixels for parks/forests and yellow pixels for desert/farmland. The remaining white pixels are then valid for station placement. An alternative to this approach would be to project the generated station locations onto roads, which would also avoid stations being placed in the middle of a building block. Furthermore, as can be seen in Figure 2, stations are also sometimes placed in tunnels or on islands (e.g., the Holland Tunnel in New York, and Liberty Island), which is also something that can be avoided. This is currently left as future work as the current filtering procedure is accurate enough for illustrative purposes.

Note that the choice to use the pre-generated tiles means that the image files include place names, which causes some distortions — for example, the names of the hilltops in the top-left corner of Figure 1a result in small patches of white

---

[2] In terms of their RGB (Red, Green, Blue) values.

pixels in the top left corner of Figure 1b. This could be remedied by rendering the tiles manually using open source software such as, for example, Mapnik[3]. Again, this is left as future research.

After filtering out the invalid areas, a further refinement would be to exclude areas that are otherwise not part of the target area. For example, the area across the river Besòs in the top-right corner of Figure 1b is not part of Barcelona but of a neighbouring city (Santa Coloma de Gramenet), and the existing BSS does not cover this area even though its city centre would have a fairly large attractiveness rating (as per Section 4.4). Hence, we restrict the target area further by instead considering the convex hull spanned by the 100m areas around the existing BSS stations (in the discretised pixel map). Without this step, several simulation techniques would do considerably worse, in particular the regular grid of Section 4.1 and the Poisson point process of Section 4.2.

3.2   Topology and Characteristics

A BSS topology can be characterised in many different ways: here, we only discuss characteristics that are purely spatial. A complete spatial characterisation of a BSS topology with N stations is a set of points

$$(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \dots, \boldsymbol{\gamma}_N) = ((\phi_1, \lambda_1), (\phi_2, \lambda_2), \dots, (\phi_N, \lambda_N)),$$

where ϕ_i is the latitude of station i and λ_i its longitude. Since we are mostly interested in these locations as projected onto a map of $I \times J$ pixels (see Section 3.1), we will also consider

$$(\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_N) = ((x_1, y_1), (x_2, y_2), \dots, (x_N, y_N))$$

where

$$x_i = \left\lfloor J \frac{\lambda_i - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right\rfloor \text{ and } y_i = \left\lfloor I \frac{\phi_i - \phi_{\min}}{\phi_{\max} - \phi_{\min}} \right\rfloor$$

We choose the centre point $\boldsymbol{c}_0 = (x_0, y_0)$ of the BSS to be the point

$$\left(\frac{1}{N} \sum_{i=1}^{N} x_i, \frac{1}{N} \sum_{j=1}^{N} y_j \right).$$

If an existing topology is not available one can alternatively take $(\lfloor J/2 \rfloor, \lfloor I/2 \rfloor)$.

Given two stations' latitude/longitude coordinates $\gamma_i = (\phi_i, \lambda_i)$ and $\gamma_j = (\phi_j, \lambda_j)$, the distance between them is given by the Haversine formula, computed as

$$d(\boldsymbol{\gamma_i}, \boldsymbol{\gamma_j}) = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos(\phi_i) \cos(\phi_j) \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right), \quad (1)$$

where $\Delta\phi = \phi_j - \phi_i$, $\Delta\lambda = \lambda_j - \lambda_i$, and R is the radius of the earth, i.e., approximately $6\,371$km.

---

[3] http://http://mapnik.org/

In Section 5, we discuss several more abstract metrics that characterise station topologies, all of which involving the distance between stations. In particular, we will consider the following two measures from [21]:

- $\bar{\delta}$: the average distance over all stations of the distance from a station to its nearest neighbour, given by

$$\delta_i = \min_{\substack{j \in \{1,\dots,N\} \\ j \neq i}} d(\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_j), \text{ and}$$

- the compactness ratio: the ratio between the area of the convex hull to that of a circle with the same circumference.

Additionally, we use the following measures:

- σ_δ: the standard deviation of the nearest neighbour distances,
- $\bar{t}$: the average edge length in the (Euclidean) minimum spanning tree of all the station locations, and
- $\max_t$: the maximum edge length in the (Euclidean) minimum spanning tree of all the station locations.

We consider the minimum spanning tree to identify BSSs that are 'disconnected' is the sense that there exist clusters of stations such that the stations within the clusters may be close together but the clusters themselves are far apart — this information is not captured by the nearest-neighbour distance δ.

3.3 Simulation Methodology

As mentioned earlier, we consider a city map consisting of $I \times J$ pixels, with I representing the height of the image file and J the width. The final number N of stations is assumed to be fixed *a priori*, and equal to the size of the city's existing BSS in Section 5. Each pixel corresponds to a (roughly) rectangular area in the city. The first step of the procedure is to assign to each pixel i, j a probability weight w_{ij}, with a higher weight corresponding to a larger probability of being selected. We also define $\forall i \in \{1, \dots, I\}, j \in \{1, \dots, J\}$, the cumulative weights

$$c_{ij} = \sum_{\substack{(k,l): \begin{subarray}{l} k < i, \text{ or} \\ k = i \wedge l \leq j \end{subarray}}} w_{kl}$$

We then generate a (pseudo-)random number u on $[0, 1)$ using (for example) Java's built-in random number generator, and find the first pixel for which $c_{ij}/c_{IJ} > u$. The centre of the rectangle corresponding to this pixel is then selected as the next station's location. We repeat this procedure until we have N stations, and allow the weights to change depending on the locations of the previously sampled stations. For specific applications, this is not necessarily the most efficient procedure, but its appeal is its generality — we apply this procedure for all procedures except the deterministic procedure of Section 4.1. In Section 4, we discuss various ways to obtain weights w_{ij}. Note that although these processes are typically treated as continuous-space processes in the literature, we evaluate their probabilities on the discrete grid of the pixel map.

4 Topology Generation Models

In this section we discuss the procedure for obtaining station configurations. We discuss four methods: the regular grid (Section 4.1), the Poisson point process (Section 4.2), the Ginibre point process (Section 4.3) and the rating-weighted Poisson process (Section 4.4). All procedures except the first can be fully described in terms of the weights w_{ij} that inform the simulation methodology of Section 3.3.

4.1 Regular Grid

The idea behind the regular grid is straightforward: we place the stations on a square or hexagonal grid in order to optimise the coverage of the target areas. In principle, this means that each station has 4 or 6 nearest neighbours, with distances δ usually the same across all stations (this is typically not true due to the presence of invalid areas such as water bodies and parks). Given a choice for a square or hexagonal grid, a BSS topology can be defined uniquely using a single value for δ if we require that a station must be placed at the centre point (x_0, y_0). The question is then how to choose δ such that there are N stations in the target area. A complication here is the fact that we want to avoid placing stations in the invalid locations discussed in Section 3.1. One possibility is an approach based on bisection: we initialise $\delta_{\max}$ to be some large value (e.g., the length of the diagonal in the bounding box of the target area) and initialise $\delta_{\min} = 0$, then check how many stations are placed within the target area for

$$\delta = \frac{1}{2}(\delta_{\max} - \delta_{\min}).$$

If this number is too large, we set $\delta_{\max} = \frac{1}{2}(\delta_{\max} - \delta_{\min})$, else set $\delta_{\min} = \frac{1}{2}(\delta_{\max} - \delta_{\min})$ and repeat until the new value of δ yields exactly N stations. One complication is that if the target area is not convex, N is not monotonously decreasing as a function of δ. Even if we use the convex hull to narrow the target area as displayed in Figure 1c, the exclusion of invalid areas such as water bodies will often result in non-convexity. Hence, this approach is approximative at best. When this is finished it is possible to add random noise (e.g., Gaussian noise) to make the topology look less artificial (we will not do this in Section 5).

4.2 Poisson Point Process

The Poisson point process is the most straightforward simulation procedure — the idea is to draw station locations uniformly within the target area. In terms of the procedure of Section 3.3, this amounts to fixing a constant $k > 0$ and setting $w_{ij} \equiv k \ \forall i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}$. In continuous space, this approach is called the Poisson point process because the ordered sequence of longitudes forms a realisation of a Poisson process conditional on it having N points within the box, and similarly for the latitudes. To avoid stations being drawn outside

the target area or in invalid locations (as discussed in Section 3.1), we set w_{ij} to zero for each point (i, j) corresponding to such a pixel. This results in a Poisson process that is *inhomogeneous* across the bounding box. In fact, the two approaches in the remainder of Section 4 can also be seen as discrete versions of inhomogeneous Poisson point processes.

4.3 Ginibre Point Process

The origins of the Ginibre point process lie in physics, where it is used to model the locations of particles in a cloud. It has recently been applied in the context of telecommunications systems, namely to model locations of base stations in a cellular network [19]. It can be defined iteratively using the following density (see, e.g., [6]):

$$f(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}\dots\boldsymbol{z}_1) = \frac{1}{k!} \cdot \frac{1}{\pi} e^{-|\boldsymbol{z}_k|^2} \cdot \prod_{j=1}^{k-1} |\boldsymbol{z}_k - \boldsymbol{z}_j|^2 \tag{2}$$

Generating a sample $(\boldsymbol{z}_1, \dots, \boldsymbol{z}_N)$ from the standard Ginibre point process is straightforward: let A and B be $N \times N$ matrices filled with realisations of random variables with a normal distribution with mean 0 and variance $\frac{1}{2}$. Then if $\lambda_1, \dots, \lambda_N$ are the eigenvalues of the complex matrix $A + iB$, then

$$\left((\mathrm{Re}(\lambda_1), \mathrm{Im}(\lambda_1)), \dots, (\mathrm{Re}(\lambda_N), \mathrm{Im}(\lambda_N)) \right) \tag{3}$$

is a realisation of the standard Ginibre point process. It is *scale-invariant*: multiplying the points of a *standard* Ginibre point process by a constant yields a Ginibre point process generated by normally distributed random variables with a different variance. Again, we do not want stations to be placed in rivers or farmland. Our approach is to use (2) to inform the weights w_{ij}, setting w_{ij} to zero for invalid pixels. An alternative approach would be to draw, using (3), a topology of size larger than N, and try to find a scale such that N stations are within the target area and in valid areas. However, this approach does not fit into our general methodology. Note that the procedure based on (2) can be computationally expensive compared to one based on (3) as the weights have to be recomputed each time a station is drawn. This is done efficiently by multiplying the elements of the current matrix by a factor that depends on the distance between each pixel and the station location sampled in the current iteration.

4.4 Rating-Weighted Scheme

The idea behind this approach is to incorporate geographical information into the weights. This is done in the following way. We use a matrix ρ_{ij} to denote the attractiveness of pixel (i, j) — this is initially set to 0. We then download a list of amenities and shops in the target area from the OSM website as discussed in

Section 2.2. This is followed by going through this list and increasing ρ_{ij} by 5 if the Euclidean distance between the location of the element of the list and the centre of the rectangle corresponding to pixel i, j is smaller than 50 meters — if this distance is between 50 and 200 meters we increase ρ_{ij} by 2, and if it is between 200 and 1000 meters we increase ρ_{ij} by 1. Of course, this is but one of many ways to incorporate the information in the list of places.

Additionally, inspired by the Ginibre point process we will also discourage stations being placed within close proximity of each other. Given that n station locations have already been drawn, let γ_{ij} be the latitude-longitude coordinate corresponding to the centre of pixel (i, j) — we then set

$$w_{ij} = \rho_{ij} \prod_{k=1}^{n} \left(1 - \exp\left(\frac{-(d(\boldsymbol{\gamma}_{ij}, \boldsymbol{\gamma}_k))^2}{\sigma} \right) \right).$$

Like the Ginibre approach, this approach can be computationally expensive: the rating needs to be computed for each pixel, and if we want to let the rating depend on the locations of the stations sampled thus far (to avoid clustering), we need to recompute it after each successful sample. Efficiency can be improved by aggregating pixels, or by using an estimate of $\rho_{\max}$ rather than the exact value.

5 Analysis & Results

In this section we will provide an overview of simulation experiments done using the methodology presented earlier. The case study will feature a selection of 11 cities. The difference between the methods and the real systems will be illustrated using several metrics, to be discussed below.

We begin with an overview of the characteristics discussed in Section 3.2 for a number of cities, presented in Table 1. We have made a selection from the close to 400 BSSs available on the `citybik.es` website, based on a number of basic metrics. First of all, the BSSs of *Barcelona*, *Paris*, *London*, and *New York* were considered because of their (large) size. Furthermore, the New York BSS has the interesting property that the station density is much higher in Manhattan and Brooklyn than in New Jersey. *Brussels* has a very compact BSS, whereas *Nice* is much more strip-like, as evidenced by the high and low compactness ratios respectively. *Melbourne* has a very disconnected BSS, with several southern stations that are very far away from the others, as evidenced by its large value for $\max_t$, the largest edge length in the minimum spanning tree. Finally, we also include *Dublin*, *Glasgow*, *Pisa* and *Tel Aviv*.

In general, there is some degree of variety in terms of the inter-spacing, as evidenced by $\bar{\delta}$ and $\bar{t}$, which are strongly correlated. For example, the average distance to the nearest other station is twice as high in Brussels as it is in London and Paris. There are also considerable differences between the ratios of σ_δ to $\bar{\delta}$, and indication of the standard deviation relative to the mean. For Brussels this ratio is very low (about $\frac{1}{3}$), and much higher for Nice (about $\frac{2}{3}$), meaning that the stations are more evenly spread out in Brussels than they are in Nice. This is also evident if one compares Figure 2a to Figure 2b.

City	# stations	$\bar{\delta}$	σ_δ	$\bar{t}$	$\max_t$	compactness
Barcelona	465	168.79	99.53	218.61	597.01	.83
Brussels	343	387.11	120.82	431.42	915.01	.93
Dublin	101	196.30	63.95	232.55	438.52	.83
Glasgow	41	420.75	222.77	512.70	1203.02	.79
London	733	213.40	90.65	253.13	700.89	.81
Melbourne	49	431.63	206.95	530.79	1905.63	.80
New York	511	247.98	117.41	283.82	1425.83	.78
Nice	176	237.20	150.86	274.16	1531.63	.55
Paris	1225	218.84	101.94	252.27	1301.27	.88
Pisa	15	644.04	376.21	708.41	1687.53	.75
Tel Aviv	196	339.49	127.04	389.96	1151.84	.73

Table 1. Comparison of the systems under consideration in terms of the characteristics discussed in Section 3.2.

In Tables 2 and 3, we compare the performance of the topology generation methods in terms of the degree to which their results match the real systems across a range of simulation experiments. In particular, this is done using two different metrics: the coverage overlap and the mean absolute difference. The former metric (the coverage overlap) is calculated as follows: given a city, a coverage radius D, and a topology generation method, we apply the following procedure in each simulation run: we create an image such that each pixel corresponding to an area within D meters of a generated BSS station is marked as 'covered' (e.g., by colouring it pink as in Figure 2). We also do that for the original BSS, and then check for how many pixels within the target area (the convex hull mentioned earlier, minus the invalid areas) the original and generated BSS give the same result. This is divided by the total number of pixels in this target area to give an overlap score. A comparison in terms of this metric is done in Table 2, for a radius D of 200m.

For the latter metric (the mean absolute difference), we compute a coverage 'score' for each pixel based on how many stations are in its vicinity. In particular, each station adds a score of 5 to all pixels whose centre is at most 50 metres away from it, a score of 2 to all pixels between 50 and 200 metres and a score of 1 to all pixels between 200 meters and 1 kilometre. The metric is then computed as the absolute difference between the scores for the real system and the generated one, averaged over all pixels (again restricted to the target area). The results in terms of this metric are displayed in Table 3.

In both tables, the numbers are the averages of 10 of such scores, plus/minus 1.96 times the standard deviation of the estimator, which gives a rough indication of the accuracy of the estimates in terms of a (rough) 95% confidence interval. Note that high scores are desirable for the coverage overlaps, whereas low values are desirable for the mean absolute differences. The methods which yielded the best results for a given city have been made bold in both tables.

For the 200m coverage overlaps, the rating-weighted scheme is nearly always the best choice, with Barcelona, Dublin, and Pisa the only exceptions.

One possible explanation for the low overlap in Dublin is that its BSS is more commuting- than leisure-oriented. The regular grid does fairly well in all cases. For the mean absolute difference, the rating-weighted method is the best choice in all cases, although in many cases (e.g., Melbourne and Pisa) the difference between the regular grid and the rating-weighted scheme is small. The rating-weighted scheme is particularly good in cities where the BSS stations are more unevenly distributed across the city, like New York and Nice. The rating-based scheme also does reasonably well for London, Paris and Tel Aviv. It should be noted that there is still substantial room for parameter tuning for some methods (particularly Ginibre).

City	Reg. Grid	Poisson	Ginibre	Rated
Barcelona	**.64**	.56 $\pm$ .01	.59 $\pm$ .01	.63 $\pm$ .01
Brussels	.54	.57 $\pm$ .01	.57 $\pm$ .01	**.60** $\pm$ **.01**
Dublin	**.61**	.54 $\pm$ .01	.55 $\pm$ .01	.59 $\pm$ .02
Glasgow	.63	.64 $\pm$ .02	.66 $\pm$ .01	**.69** $\pm$ **.02**
London	.56	.52 $\pm$ .01	.48 $\pm$ .01	**.59** $\pm$ **.01**
Melbourne	.62	.64 $\pm$ .02	.64 $\pm$ .01	**.68** $\pm$ **.01**
New York	.53	.52 $\pm$ .01	.49 $\pm$ .01	**.67** $\pm$ **.01**
Nice	.52	.53 $\pm$ .02	.56 $\pm$ .01	**.67** $\pm$ **.02**
Paris	.62	.55 $\pm$ .01	.56 $\pm$ .01	**.66** $\pm$ **.01**
Pisa	**.73**	.72 $\pm$ .02	.72 $\pm$ .02	.72 $\pm$ .03
Tel Aviv	.54	.55 $\pm$ .01	.58 $\pm$ .01	**.61** $\pm$ **.01**

Table 2. Comparison of the topology generation methodologies in terms of coverage overlap as discussed in Section 5. High values indicate a large overlap — the best-performing methods are marked in bold for each city.

City	Reg. Grid	Poisson	Ginibre	Rated
Barcelona	17.55	18.98 $\pm$ .84	24.51 $\pm$ .08	**12.29** $\pm$ **.47**
Brussels	5.88	7.08 $\pm$ .33	6.92 $\pm$ .06	**5.38** $\pm$ **.24**
Dublin	9.51	11.07 $\pm$ 1.68	15.69 $\pm$ .27	**9.06** $\pm$ **.43**
Glasgow	4.54	5.71 $\pm$ .75	4.43 $\pm$ .16	**3.23** $\pm$ **.24**
London	14.39	15.02 $\pm$ .39	24.57 $\pm$ .03	**9.59** $\pm$ **.25**
Melbourne	3.69	4.57 $\pm$ .33	4.82 $\pm$ .08	**3.67** $\pm$ **.29**
New York	14.39	14.83 $\pm$ .54	20.00 $\pm$ .04	**5.44** $\pm$ **.15**
Nice	11.75	13.04 $\pm$ .49	18.63 $\pm$ .07	**6.55** $\pm$ **.29**
Paris	19.03	19.42 $\pm$ .54	25.69 $\pm$ .04	**13.05** $\pm$ **.33**
Pisa	3.03	3.59 $\pm$ .45	4.18 $\pm$ .20	**2.93** $\pm$ **.38**
Tel Aviv	7.71	8.26 $\pm$ .42	10.67 $\pm$ .06	**5.43** $\pm$ **.22**

Table 3. Comparison of the topology generation methodologies in terms of the mean absolute difference as discussed in Section 5. Low values indicate a small difference — the best-performing methods are marked in bold for each city.

6   Conclusions

In this paper, we have introduced a new approach for generating station topologies. Of the four methods proposed, the one whose generated topologies matched the existing systems in the best manner was the data-driven approach, in which areas of the city were 'rated' according to popularity.

In future work, we are planning to compare the simulation results to 'optimal' results obtained by the deterministic optimisation methods discussed in Section 2.1, e.g., the ones presented in [10]. We are also planning to consider a broader variety of rating methods: e.g., closeness to railway stations for a commuting-oriented BSS, or greater weight to landmarks for a tourism-oriented system. Additionally, we hope to link our methodology to bike movement models: classifying stations in terms of their typical usage behaviour (e.g., during peak hours or weekends) based on their location. The programming code used for the experiments will be posted online in the near future. Finally, there are some minor adjustments mentioned in the texts: e.g., generating bespoke OSM tiles using Mapnik, or projecting station locations onto roads.

Acknowledgments

References

1. *CityBikes API.* `http://api.citybik.es/v2/`. Accessed: 2015-01-28.
2. Shitai Bao, Ningchuan Xiao, Zehui Lai, Heyuan Zhang & Changjoo Kim (2015): *Optimizing watchtower locations for forest fire monitoring using location models.* Fire Safety Journal 71, pp. 100–109.
3. Longbiao Chen, Daqing Zhang, Gang Pan, Xiaojuan Ma, Dingqi Yang, Kostadin Kushlev, Wangsheng Zhang & Shijian Li (2015): *Bike sharing station placement leveraging heterogeneous urban open data.* In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ACM, pp. 571–575.
4. Etienne Côme & Latifa Oukhellou (2014): *Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Vélib' System of Paris.* ACM Transactions on Intelligent Systems and Technology (TIST) 5(3), p. 39.
5. Edoardo Croci & Davide Rossi (2014): *Optimizing the Position of Bike Sharing Stations. The Milan Case.*
6. Laurent Decreusefond, Ian Flint & Anaïs Vergne (2013): *Efficient simulation of the Ginibre point process.* arXiv preprint arXiv:1310.0800.
7. Elliot Fishman (2015): *Bikeshare: A Review of Recent Literature.* Transport Reviews (ahead-of-print), pp. 1–22.
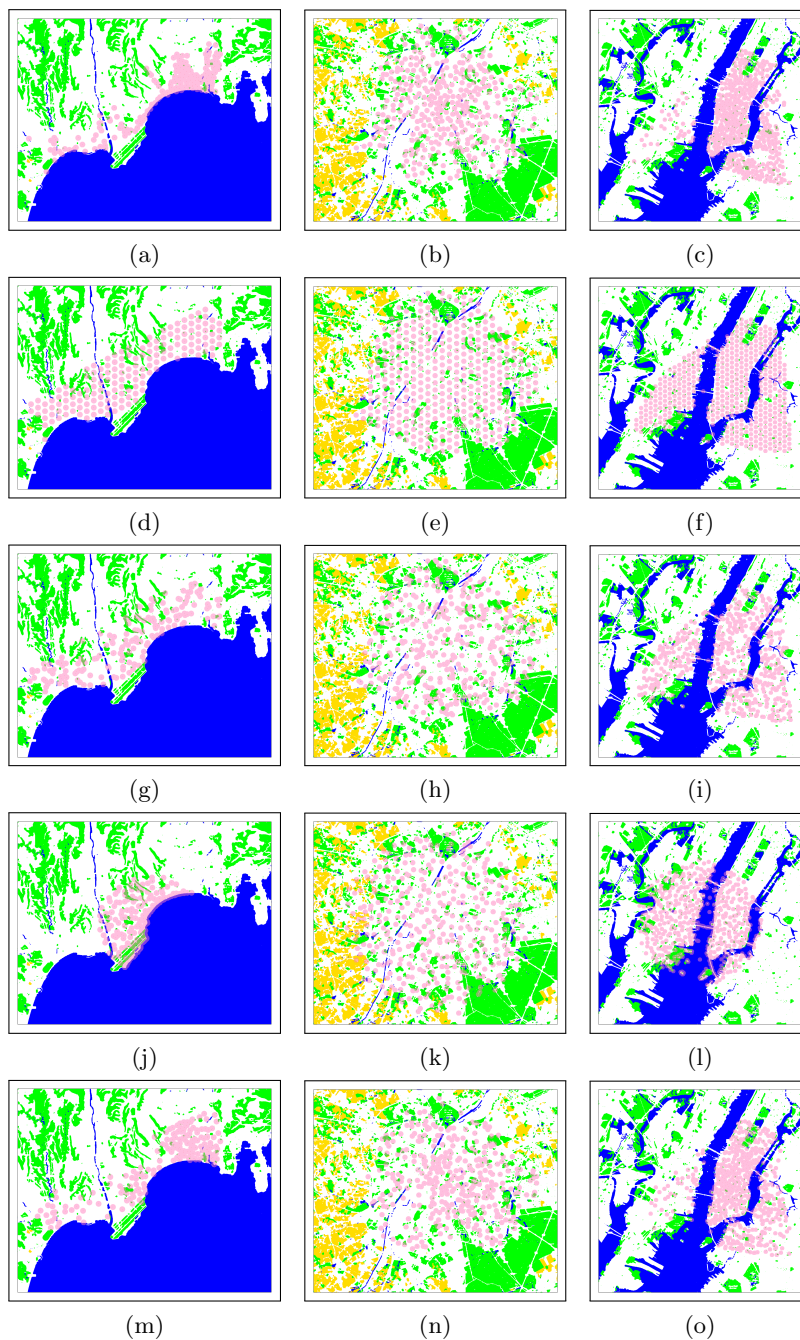
Fig. 2. Left to right: Nice, Brussels, New York. Top to bottom: real system, regular grid (without noise), Poisson, Ginibre, rating-weighted.

8. Ines Frade & Anabela Ribeiro (2015): *Bike-sharing stations: A maximal covering location approach*. Transportation Research Part A: Policy and Practice 82, pp. 216–227.

9. Jon Froehlich, Joachim Neumann & Nuria Oliver (2009): *Sensing and Predicting the Pulse of the City through Shared Bicycling*. In: IJCAI, 9, pp. 1420–1426.

10. Juan Carlos García-Palomares, Javier Gutiérrez & Marta Latorre (2012): *Optimizing the location of stations in bike-sharing programs: a GIS approach*. Applied Geography 35(1), pp. 235–246.

11. Nicolas Gast, Guillaume Massonnet, Daniël Reijsbergen & Mirco Tribastone (2015): *Probabilistic Forecasts of Bike-Sharing Systems for Journey Planning*. In: The 24th ACM International Conference on Information and Knowledge Management (CIKM 2015).

12. Ebru Aydin Gol, Ezio Bartocci & Calin Belta (2014): *A formal methods approach to pattern synthesis in reaction diffusion systems*. In: 2014 IEEE 53rd Annual Conference on Decision and Control, IEEE, pp. 108–113.

13. Marcel C Guenther & Jeremy T Bradley (2013): *Journey data based arrival forecasting for bicycle hire schemes*. In: Analytical and Stochastic Modeling Techniques and Applications, Springer, pp. 214–231.

14. Mordechai Haklay & Patrick Weber (2008): *Openstreetmap: User-generated street maps*. Pervasive Computing, IEEE 7(4), pp. 12–18.

15. Sakari Jäppinen, Tuuli Toivonen & Maria Salonen (2013): *Modelling the potential effect of shared bicycles on public transport travel times in Greater Helsinki: An open data approach*. Applied Geography 43, pp. 13–24.

16. Neal Lathia, Saniul Ahmed & Licia Capra (2012): *Measuring the impact of opening the London shared bicycle scheme to casual users*. Transportation research part C: emerging technologies 22, pp. 88–102.

17. Junming Liu, Qiao Li, Meng Qu, Weiwei Chen, Jingyuan Yang, Xiong Hui, Hao Zhong & Yanjie Fu: *Station Site Optimization in Bike Sharing Systems*.

18. Russell Meddin & Paul DeMaio: *The bike-sharing world map*. `https://www.google.com/maps/d/viewer?mid=zGPlSU9zZvZw.kmqv_ul1MfkI`. Accessed: 2015-01-28.

19. Naoto Miyoshi, Tomoyuki Shirai et al. (2014): *A cellular network model with Ginibre configured base stations*. Advances in Applied Probability 46(3), pp. 832–845.

20. Rahul Nair, Elise Miller-Hooks, Robert C Hampshire & Ana Bušić (2013): *Large-scale vehicle sharing systems: analysis of Vélib'*. International Journal of Sustainable Transportation 7(1), pp. 85–106.

21. Oliver O'Brien, James Cheshire & Michael Batty (2014): *Mining bicycle sharing data for generating insights into sustainable transport systems*. Journal of Transport Geography 34, pp. 262–273.

22. Yu-Shiuan Tsai, Patrick Chow-In Ko, Chung-Yuan Huang & Tzai-Hung Wen (2012): *Optimizing locations for the installation of automated external defibrillators (AEDs) in urban public streets through the use of spatial and temporal weighting schemes*. Applied Geography 35(1), pp. 394–404.

23. Wei Tu, Qingquan Li, Zhixiang Fang, Shih-lung Shaw, Baoding Zhou & Xiaomeng Chang (2015): *Optimizing the locations of electric taxi charging stations: A spatial–temporal demand coverage approach*. Transportation Research Part C: Emerging Technologies.

24. Ying Zhang, Mark Zuidgeest, Mark Brussel, Richard Sliuzas & Martin van Maarseveen (2013): *Spatial location-allocation modeling of bike sharing systems: a literature search*. In: Proceedings of the 13th World Conference on Transportation Research.