QUANTICOL

A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours

quanticol

http://www.quanticol.eu

D2.2

From spatial data to spatial models

Revision: 1.0

Author(s): Vashti Galpin (UEDIN), Cheng Feng (UEDIN), Daniël Reijsbergen (UEDIN)

Due date of deliverable: Month 36 (March 2016)

Actual submission date: Mar 31, 2016

Nature: R. **Dissemination level:** PU

Funding Scheme: Small or medium scale focused research project (STREP)
Topic: ICT-2011 9.10: FET-Proactive 'Fundamentals of Collective Adaptive Systems' (FOCAS)
Project number: 600708

Coordinator: Jane Hillston (UEDIN)
e-mail: `Jane.Hillston@ed.ac.uk`
Fax: +44 131 651 1426

Part. no.	Participant organisation name	Acronym	Country
1 (Coord.)	University of Edinburgh	UEDIN	UK
2	Consiglio Nazionale delle Ricerche – Istituto di Scienza e Tecnologie della Informazione "A. Faedo"	CNR	Italy
3	Ludwig-Maximilians-Universität München	LMU	Germany
4	Ecole Polytechnique Fédérale de Lausanne	EPFL	Switzerland
5	IMT Lucca	IMT	Italy
6	University of Southampton	SOTON	UK
7	Institut National de Recherche en Informatique et en Automatique	INRIA	France

COOPERATION

Executive summary

This deliverable describes the research into the use of spatial data for spatial modelling that has taken place within the QUANTICOL project. Space plays an important role in QUANTICOL because of the nature of the case studies we have chosen as examples of collective adaptive systems. When considering smart transport, location in space is usually relevant to the behaviour and dynamics of the system. In the case of smart grids, location may play less of a role in the distribution of energy but in certain scenarios such as residential smart grids, there can be a spatial element to models. There are two distinct approaches to take with spatial data: to abstract away from it or to use it. This document focusses on the latter, and considers how spatial data can be used for model construction and parameter estimation.

Many public transport systems collect GPS data of vehicle locations. We have developed a methodology for taking such vehicle data and developing parameterised patch-based (discrete-space) models of vehicle routes. The methodology includes a phase of patch identification once automated map generation has been used to identify routes. GPS data can then be used to derive parameters for the time taken to cross a patch for hyper-Erlang and shifted Erlang distributions. The models developed and parameterised with this approach have been used to assess the impact of speed limit reductions and for statistical model checking of properties, such as those describing punctuality.

Space can also be considered in a static manner, and we have characterised the spatial distribution of various bike sharing schemes. Furthermore, we have investigated which distributions can be used to generate topologies of bike stations that are closest to the actual arrangement for a number of cities. The approaches under consideration are the regular grid, the Poisson spatial point process, the Ginibre spatial point process, and a placement based on rating of amenities. Using street map data to exclude areas where stations cannot be placed, pixels are weighted as to their likelihood of being the position of a bike station. Two different approaches to the assessment of the coverage of the generated topologies when compared with the actual topologies are presented.

Journey data can also be used in model development and parameterisation. This data can be synthetic or real. An example of the use of real journey data from a bike-sharing scheme is in determining which bike stations have an influence on other stations, thus capturing the spatial layout and interactions of these stations. By considering only the stations with a high level of influence, the model can be reduced in size and thus its analysis becomes possible. Synthetic journey data has been used in the transformation of an individual continuous-space model to a population discrete-space model using patches. Here, a description of movement was used to create a small simulation of individual movement, and from this parameters were estimated for the population model.

This deliverable also considers how the research involving spatial data relates to the other work packages in the QUANTICOL project, including work on case studies, the Carma modelling language and spatial logics.

Contents

1 Introduction

In many collective adaptive systems (CAS), space plays an important role. This is particularly true of the smart transport case studies on which we focus in the QUANTICOL project, and to some extent, true of the smart grid case studies. In Task 2.3 of Work Package 2, the focus is on the use of spatial data in constructing models that consider space explicitly. This covers how spatial data can influence structure of models and also how it can be used to inform the parameters of models. A particular technique is parameter estimation which involves taking data from the real world and using this data to estimate parameters for a specific modelling abstraction of that real world artifact or system, or of the environment in which the system operates. Much research has been done into the different aspects of parameter estimation.

In some models in the QUANTICOL project such as the modelling of residential smart grids [Gal15], parameter collection is used rather than parameter estimation. For this model, there is substantial information about consumption of energy by household appliances and general consumption patterns over different times of the day, as well as specification of the renewable energy harvested by various approaches. This means that the model can be constructed with those parameters, and the behaviour of the model can then be observed. Since there is no available data for the hypothesised sharing of energy between households, there is no opportunity for parameter estimation or validation.

When considering model construction where there is location or spatial data, there are two approaches: one is to use the data to determine model structure or parameters, and the other approach is to try to abstract from it. Before going on to discuss the former in detail, we consider the latter.

In the case of prediction for bike sharing schemes, one approach is for each station to be treated in isolation from the others [GMRT15]. This treatment uses an assumption that the state of a particular station does not depend on the state of the other ones. In practice, this is not true; however the large numbers of bike stations in schemes such as the Paris *Vélib'* scheme means that it is possible to use the asymptotic analysis techniques introduced in [FGM12], which show that this isolation approximation becomes exact as the number of stations goes to infinity. We will discuss a different approach to this prediction that does take account of spatial information later in this document.

Returning now to scenarios where we do not wish to abstract from space, we present a number of different approaches to parameter estimation using spatial data developed within the QUANTICOL project, where space plays an essential role. One technique focusses on the use of individual GPS data from vehicles, and relates to a dynamic model of transport. Another uses static spatial location data to explain the location of bike stations in a bike-sharing scheme, using spatial point processes over landscape features. We also consider how journey data can be used to reduce the size of a model so that parameter fitting becomes feasible. Additionally, we describe a general technique that can be applied to journey data to create a population discrete-space model from real world journey data or generated from an agent-based continuous-space model as shown in [Fen14].

These sections are followed by conclusions where we highlight how these techniques link to the research of other work packages in the project.

2 Methodology for parameter estimation from GPS data

Modern GPS and communication technologies allow for a level of monitoring of vehicles that was previously unimaginable. At the same time computational power, combined with advances in the field of formal verification techniques, has enabled more intricate and complete system models to be analysed in a rigorous manner. There is synergy between these two developments in the sense that the ability to handle more complex models is largely futile without sufficient data for realistic parameterisation. Hence, it is vital to have a clear methodology for translating the wealth of data generated by IT infrastructures into properly parameterised formal models that are suited for modern analysis tools.

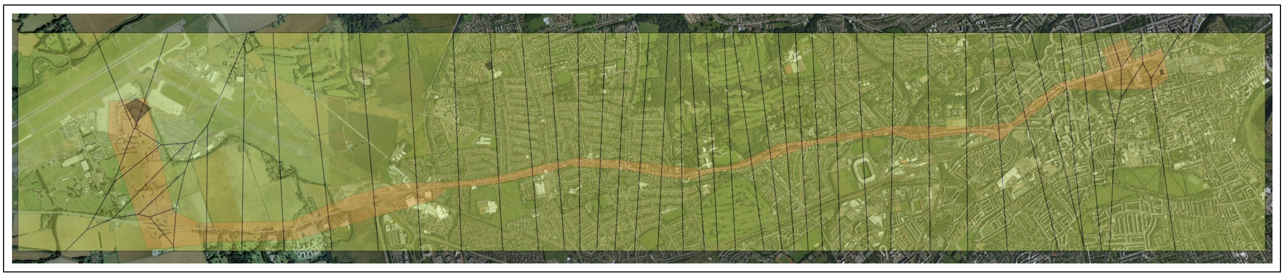
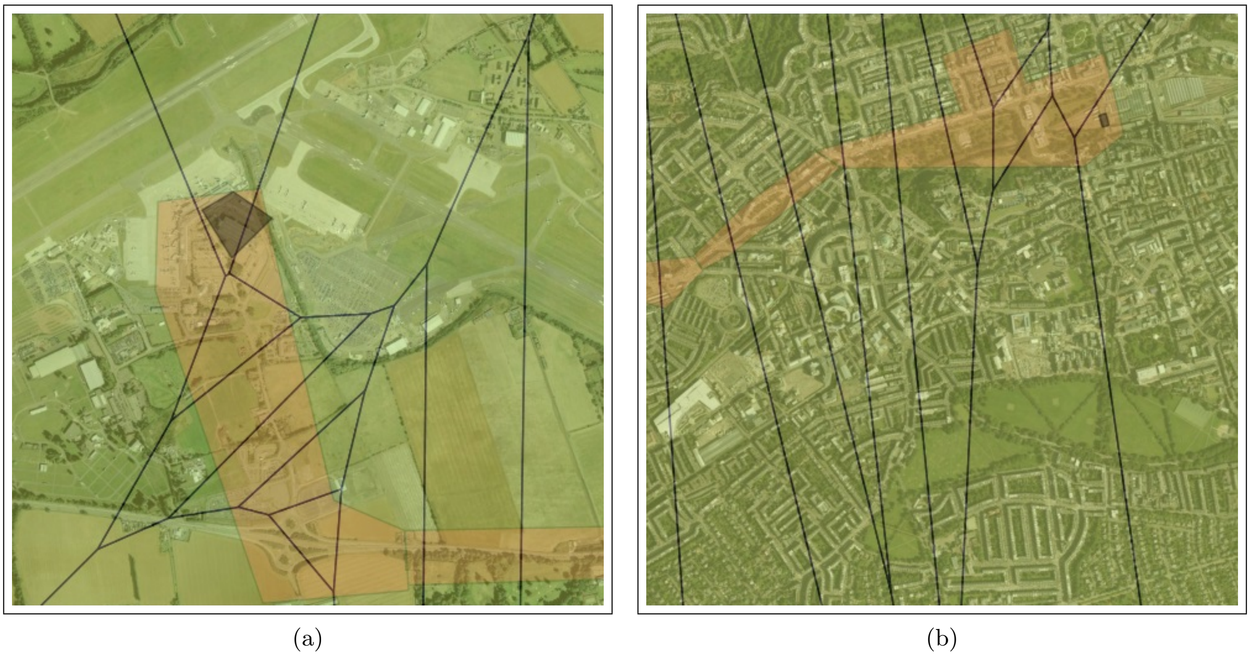


Figure 1: Patch structure based on a Voronoi tessellation generated using the K-means clustering for the Airlink route in Edinburgh. The dark lines indicate the borders between the areas of points closest to the K centre points identified by the clustering algorithm. The orange area in the middle represents the route itself. In the west, the bus is able to drive faster which means that there will be fewer bus GPS measurements, hence the cluster centres will be further apart than in the more congested areas. In Figure 2, we zoom in on the ends of the route.

This section reports on a formal methodology to enable the construction, including parameterisation, of stochastic public transportation models. The modelling approach is *patch-based* and the transport routes are partitioned into patches, meaning that vehicles move from patch to patch such that the time spent in each patch is modelled using a formally defined probability distribution. Both the patch identification and the parameterisation of the probability distributions are done using real-world Automatic Vehicle Location (AVL) data. We use the modelling formalism of *probabilistic timed automata* to describe our models, as this gives us access to the powerful model checking tool UPPAAL [BDL04]. Model checking is a formal verification technique in which the veracity of a clearly specified *property* is evaluated. Examples of the type of properties of interest involve requirements on *punctuality* as often specified by regulatory bodies of public transport. The code used to conduct the experiments presented in this document is open-source and available online.



(a)

(b)

Figure 2: Close-up of the western and eastern ends of Figure 1. Again, the dark lines represent the borders between the patches. For some patches, the distance travelled by the Airlink route within it will vary significantly depending on the direction.

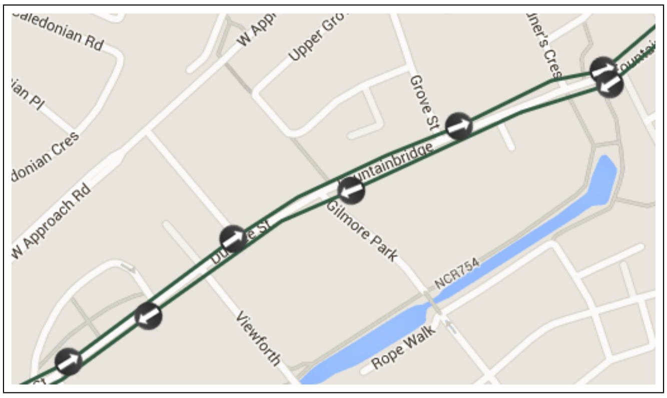


Figure 3: There is not always a clear one-to-one correspondence between the stops serving the two directions of service 1.

This methodology draws on research conducted earlier in the QUANTICOL project [RGH15, LCG15, GR15, GTV14, RR15] and it will be reported in [RG16].

2.1 Patch identification

We discuss four ways of creating a patch structure in this section (each will be discussed in more detail below), based on the following.

1. K-means clustering.
2. Use of bus stop locations.
3. Use of service overlaps.
4. Jenks natural breaks optimisation

For approach #1, we take the full dataset of GPS measurements expressed only in terms of latitude/longitude coordinates. We then use K-means clustering to identify K centre points of the clusters. A patch structure is then obtained using the Voronoi tessellation resulting from these K centre points. An advantage of this approach is that it is easy to apply – many tools support K-means clustering (e.g., the statistical package R). However, this approach is relatively crude and cannot handle 'complex' route shapes, i.e., those that do not resemble a straight line. For example, in Figure 1, the approach works reasonably well in the middle of the route, but less so near the far ends (e.g., in the right-hand section corresponding to the city centre; see Figure 2b for a more detailed version.).

For approach #2, we need a list of bus stops and their latitude/longitude coordinates. These locations can be inferred via the method of Biagioni et al. [BGME11] (the principles of which will be discussed in Section 2.2) or obtained via the Lothian Buses API. The automated inference method of [BGME11] is not very good at distinguishing bus stops from busy junctions, although that is not necessarily an issue for patch identification purposes. Given the bus stop list, we use the Voronoi tessellation defined by the bus stops as a first patch structure, possibly augmented with separate small patches with a fixed diameter around the bus stops to isolate the large amount of time buses spend at the stops. Compared to K-means clustering, this method is still relatively easy to apply. However, although it is more refined than K-means clustering, it is non-trivial to classify bus stops based on the direction of the route that they serve, and the approach might result in many spurious patches for each of the two directions (e.g., in Figure 3, the patch structure for the west-east route would be unnecessarily impacted by the bus stops serving the east-west route).

Approach #3 particularly makes sense in a setting where we have data for multiple routes. We assume that we have knowledge of all possible routes in terms of a directed graph – routes are then

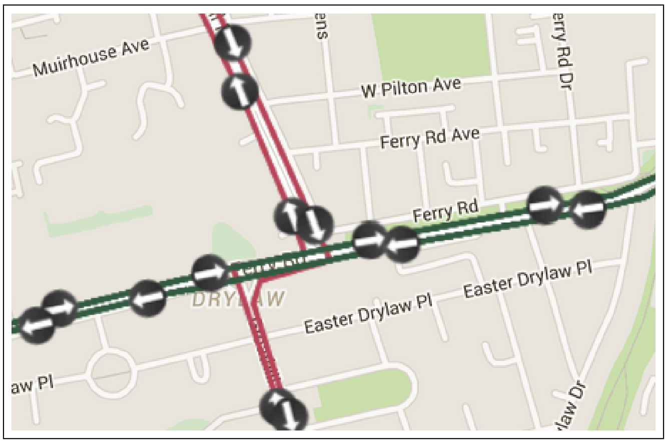


Figure 4: The part of road on which services 21 (dark green lines) and 24 (red lines) intersect is very small, resulting in spurious patches if one were to use approach #3.

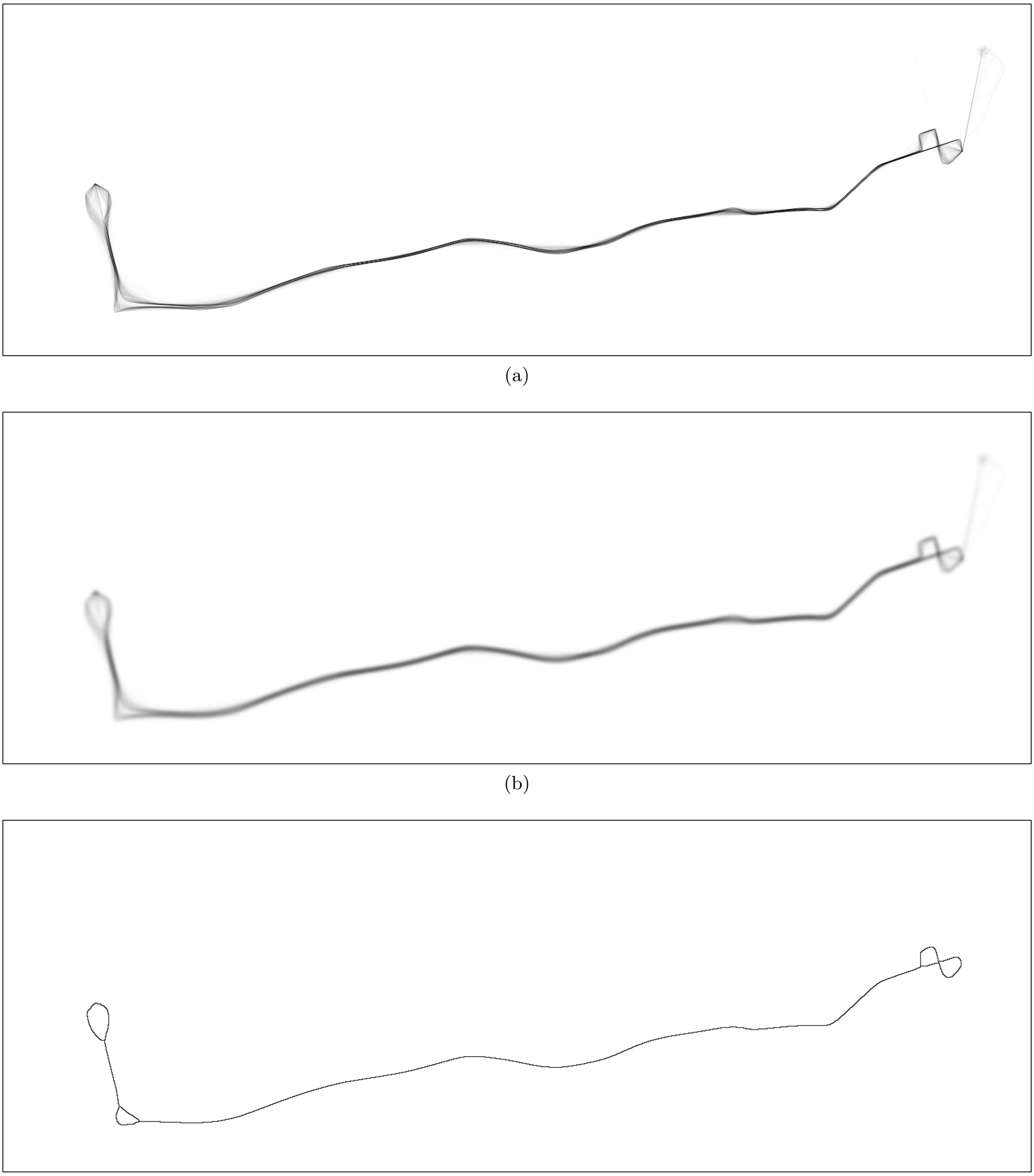
given by sequences of edges in this graph. Grouping together adjacent edges that are serviced by the same routes would be a way of obtaining a patch structure. This approach is particularly suited for modelling contention between buses of different services for road (and perhaps bus stop) capacity. However, the size differences between the patches could be very large – see, for example, Figure 4.

Finally, for approach #4, we first compute for each bus GPS measurement the degree to which the bus has completed the route at the time of measurement, as a number in $[0, 1]$. We then construct an initial patch structure by dividing the interval $[0, 1]$ into evenly-sized portions. We want to group together patches that are next to each other and are 'similar' in terms of the number of times buses were observed in the patches. There are several ways to do this – we have investigated two in more detail. The first method is to look for the pair of adjacent patches such that the merger of the two has the lowest number of bus observations. Merge these, and continue carrying out this step until the resulting patch has an observation count larger than the highest number of observations amongst the initial patches. As an advantage, this approach does not require that the final number of patches is specified *a priori*. For the second approach, we create a sequence of absolute differences between the number of observations in each patch and the next. We then use K-means clustering in one dimension (this is called Jenks natural breaks optimisation [Jen67]) to create patches.

2.2 Automated map generation

In order to perform Jenks natural breaks optimisation [Jen67], we need to be able to compute for each bus measurement the route completion degree, as a number in $[0, 1]$. This is not entirely trivial: it requires that the bus direction is known, namely that we know what distance the bus has travelled since starting its route, and the total length of the route. To do this, we first aim to represent the route as a graph, in which edges correspond to frequently travelled roads in the city. Such a graph can again be inferred via the method of Biagioni et al. [BGME11] or obtained via the Lothian Buses API (in the case of Edinburgh). Using the first approach, we would first create a heat map of GPS observations as displayed in Figure 5a, including interpolation between subsequent measurements to compensate for GPS shadows. We then apply a Gaussian blur, as displayed in Figure 5b, followed by a filter in which pixels that are lighter than a certain threshold are made completely white, and then find the skeleton version of the remaining non-white pixels, as displayed in Figure 5c.

Having obtained the skeleton map, we turn this into a graph as follows. First, we determine the locations of all the ends and crossings (specifically, all non-white pixels that do not have 0 or 2 non-white pixels in their Von Neumann neighbourhood). We then determine the lines of non-white pixels connecting these ends/crossings as pixel sequences. For each line, an initial graph representation is one where each pixel has a vertex and edges are between those pixels that are Von Neumann neighbours.



(a)

(b)

(c)

Figure 5: Automated route map for the Airlink service in Edinburgh, operated by Lothian Buses. Top figure: observation frequency map, including interpolation. Middle figure: same as the first but after the application of a Gaussian blur. Bottom figure: skeleton map obtained from the middle figure.

This graph is then pruned using the Ramer-Douglas-Peucker algorithm [DP73]. The final graph is then constructed by merging the vertices corresponding to the crossing pixels, completing the algorithm described in [BGME11].

In the resulting graph, the route can be represented by a sequence of directed edges. To determine

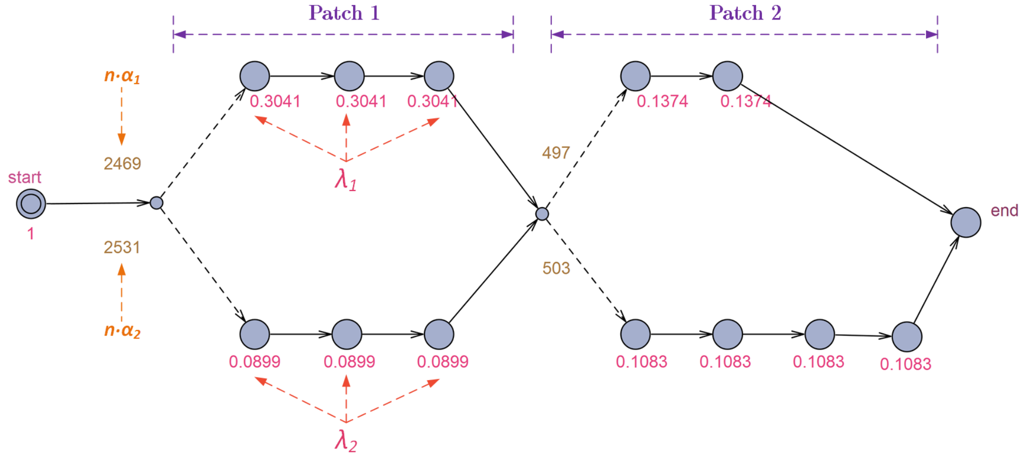


Figure 6: Illustration of a PTA representing a system with two patches with two-branch Hyper-Erlang distributed crossing times.

the route, we first identify the start and end points - we assume that those are the edges in which a large amount of time is spent. We calculate for each (directed) edge how much time is spent on it on average, divided by the length (in meters) of the edge. If this value is very high, then this is seen as an indication of the presence of a stop, start or end point. In particular, if this value for an edge is over ten times the median, then this edge is taken as a start/end point. Next, we determine the edge sequences that occur most often for buses travelling between the start/end points – these are the sub-routes, e.g., the Edinburgh Airlink has two sub-routes, namely the West-East direction and the East-West direction. The length of the total route starting from a specific start point assumed to be 'the' start point (it does not matter which one is chosen) is then the sum of the sub-routes leading back to the start point.

When this has been done, the route completion can be computed as follows: given a GPS measurement, we find the edge it is closest to and which direction the bus is going (this is learned from the previous measurements and kept track of when going through the chronologically ordered data file). This should then uniquely map to a part of the route – buses are assumed to cross each directed edge at most once during the route. The distance to the start point can be found through the edge distances, and when this is divided by the total route length, this results in the required route completion measurement.

2.3 Parameterisation of a UPPAAL Model

We have so far discussed how to obtain patches and a graph map including the start/end point of sub-routes of the service under consideration. A probabilistic model for buses moving through the patches can then be obtained using the approach presented in [RR15]. Using this approach, we build a model that represents the continuous-time movement of buses as they move along predefined segments of a route. The time taken to traverse a patch is modelled as a random variable whose distribution is chosen to best fit the GPS data provided to us by Lothian Buses. In [RR15], two classes of distributions are used to model the time in each of the patches. The first of these is the class of hyper-Erlang distributions, a modelling class with good performance in a broad range of settings and which is easily parameterised using the user-friendly tool HyperStar [RKW12]. The second is a variation of the three-parameter gamma distribution, which is an industry standard recommended by the Traffic Engineering Handbook [Pli92]; its parameters are determined using a purpose-built optimisation tool written in the language of the statistical package R [R D05]. The second distribution will be called 'shifted Erlang' in this document. The resulting distributions are then used to build probabilistic models in the tool

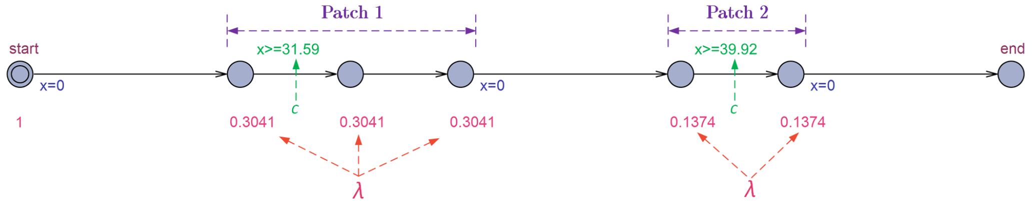


Figure 7: Illustration of a PTA representing a system with two patches with shifted Erlang distributed crossing times.

UPPAAL [BDL04], which facilitates the modelling formalism of Probabilistic Timed Automata and the simulation-based verification framework of Statistical Model Checking (SMC) [DLL$^+$11].

The modelling class used by UPPAAL is that of *timed automata*. Of particular interest to this paper is the UPPAAL Statistical Model Checker (SMC) [DLL$^+$11], which allows for the verification of Probabilistic Timed Automata (PTAs), the stochastic extension of timed automata. We do not give a full description of PTAs but discuss the key language features used in the document through a discussion of the PTA models for the patch-based bus model.

In Figure 6, we depict a sequence of two patches for each of which the journey time is modelled using a two-branch hyper-Erlang distribution. States (called 'locations' in PTA terminology) are represented using circles, and edges (representing transitions between states) using arrows. The system starts in the left-most location (the double circle), which is exited with a rate of 1 (this transition would ideally be immediate, but the, on average, 1-second delay is negligible). We need this state because the system requires the specification of a unique initial state. When the initial state is left, a probabilistic choice is made between the two branches of the next patch. The edge weights, which determine the relative likelihood of choosing an edge, are required to be integers. Hence, the values of α_1 and α_2 – found to be 0.4938 and 0.5062 by the parameter-fitting algorithm – had to be converted to the smallest pair of integers with the same ratio, which in this case were 2469 and 2531.

If branch i has been chosen, the system then needs to complete k_i locations that each have rate λ_i. When this has been completed, the system jumps to a so-called branch point (the small circle) from which a probabilistic choice is made between the branches of the next patch. When the final patch has been completed, the system jumps to the final state labelled 'end'.

In Figure 7, we depict a similar model for the shifted Erlang distribution. In addition to the discrete states, we now also use a real-valued clock, called x. This clock is reset immediately after a patch is left. We then impose a guard on the first (or only) edge of the patch requiring that this edge cannot be completed before the clock x has reached a value greater than c; in practice, this adds a constant delay of c to the global clock. Otherwise, the behaviour in each patch is similar to that of a single Erlang branch in Figure 6.

Whereas Figures 6 and 7 depict graphical representations of the PTA models used in this paper, the same models can also be expressed in XML format. These XML files can be generated by R, taking as input either a file with the hyper-Erlang parameters obtained via HyperStar, or a matrix of the shifted Erlang distribution parameters obtained by another function in R. Furthermore, the system property of interest is also embedded in the XML file. In [RR15], the focus was on two probabilities: the property of a bus reaching the final location under one minute early or less than five minutes late. These probabilities are inspired by the notion of a bus being 'on-time' if it is at most N_1 minutes early and at most N_2 minutes late. This is a common notion, and the particular choice of $N_1 = 1$ and $N_2 = 5$ is from the Scottish Government's Bus Punctuality Improvement Partnerships guiding document [Sg09]. For example, if the timetabled journey duration for an Airlink bus to get from Waverley station in Edinburgh's city centre to the airport is 28 minutes (i.e., 1680 seconds) then the

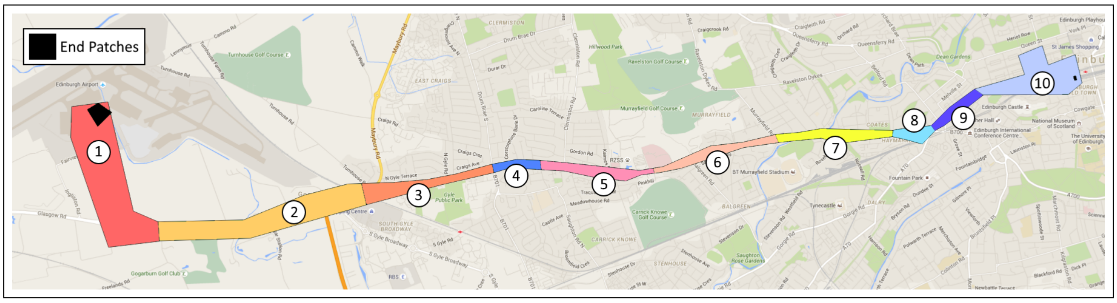


Figure 8: Illustration of the (manually assigned) patches used to partition the Airlink route. Patches 4, 8, 9, and 10 were of specific interest as they are affected by a new speed limit.

property

> ***Pr[<=1620] (<>Process.end)***

specifies the property of being over a minute early ('Process' is the name of the automaton with all the patches and 'end' the name of the final state). The SMC engine of UPPAAL is then invoked to estimate the probability of this property being satisfied. In particular, the system is repeatedly executed and the results used to make statistically justified statements about the relevant probabilities. Note that a wider range of properties can be evaluated using model checking (see [RG14] for an overview of punctuality-related probabilities for frequent bus services), although these will not be discussed further in this document.

2.4 Results

In [RR15], the parameter fitting procedure described in the previous section was applied to a real-world case study involving the introduction of a 20mph speed limit in Edinburgh. Although the patch assignment was done manually in this setting (this was done to separate the areas of the route that are affected by the new speed limit from those that are not), the same methodology can be applied using automatically generated patches. The patch structure is displayed in Fig 8 – the studies route is the Airlink service which connects Edinburgh airport to the city centre.

To give some more detail about the results of the parameter fitting procedure, we display in Table 1 for both the shifted Erlang and various hyper-Erlang distributions, the expected number of phases (the locations in the UPPAAL model discussed in Section 2.3). The addition of the shift constant typically means that distributions with fewer phases are found to be optimal (in terms of the log-likelihood

			hyper-Erlang	
Patch	shifted Erlang	Erlang	2-branch	3-branch
1	4	80	149.5	591.4
2	1	3	186.7	310.8
3	13	32	59.0	569.9
4	3	10	36.4	39.9
5	3	13	47.5	89.0
6	2	38	125.5	161.6
7	3	23	79.6	91.4
8	5	7	13.2	21.8
9	9	9	22.0	272.9
10	6	12	13.5	27.8

Table 1: The expected number of exponential phases for each of the distributions.

– see [RR15] for a more in-depth discussion). The consequences of differences in the numbers of phases are twofold. Firstly, since the time to draw a sample using the SMC engine of UPPAAL scales roughly linearly with the expected number of completed phases, it takes less time to obtain accurate estimates of the on-time probabilities using the shifted Erlang. Secondly, high phase counts can be a symptom of overfitting: it could indicate that a sharp peak around a cluster of observations has been created through an Erlang branch with many phases and a rate such that the branch mean is near the centre of the cluster. Otherwise, we found the tail behaviour of the hyper-Erlang distribution is better able to represent total journey durations, whereas the shifted Erlang is more efficient to simulate and its parameterisation is more stable across repeated experiments. Another advantage of the shifted Erlang is that the parameter fitting procedure is easy to automate, avoiding manual calls to the tool HyperStar. In the end, there is no clearly superior distribution and the best choice of modelling distribution depends on the context.

3 Spatial analysis of bike sharing station location

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 [Fis15] in 2010 to almost 1,000 at the time of writing [MD]. Hence, the question of choosing station locations in a new BSS is of increasing relevance to planners, operators, and scientists. In the literature, the best station locations are typically chosen using some form of deterministic optimisation where the optimisation criterion depends on the context, e.g., coverage of target areas. For example, the methods implemented in the geographic analysis tool ArcGIS [GPGL12] or the optimisation tool suite XPRESS [FR15].

We take a different approach and consider the use of stochastic simulation to generate BSS station locations. The probabilities informing the simulation procedure are inspired by the literature on spatial point processes. We use 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. 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. Furthermore, it captures the inherent randomness in the fact that the optimality criteria chosen to inform a deterministic optimisation procedure do not correspond 100% with the estimations of planners and operators. This research will be reported in [Rei16].

Current BSSs combat the theft and vandalism that plagued the previous ones 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, is made available to third-party users, including researchers. This data permits the development of algorithms for the positioning of station locations [GPGL12, ZZB$^+$13], which can be validated against real-world systems using the station location coordinates that are often provided alongside bike availability data. A comprehensive comparison of 38 BSSs worldwide is presented in [OCB14] in which some of the metrics deal with the geographic position of the stations and these will be considered here. Potential and existing BSSs have been studied in various cities around the world; for example, see [FNO09, LAC12, EL14, GPGL12, CZP$^+$15, JTS13, FR15, LLQ$^+$, CR14].

3.1 Data

Two main sources of data are used in this research. OpenStreetMap (OSM) provides a dataset of roads and places that forms the basis of a non-proprietary map of the world. We use the OSM database for two purposes. First of all, we use it to identify places that are unsuited for BSS station placement, and

[1] Alternatively called Bicycle-Sharing Plans.

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 2: Overview of the systems under consideration

second, we use it to collect data about places of interest. To obtain data about the station locations of existing BSSs, we use the API for the website `citybik.es`.

3.2 BSS topology characterisation

Each BSS can be characterised by a number of different measures including the following two measures from [OCB14]:

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

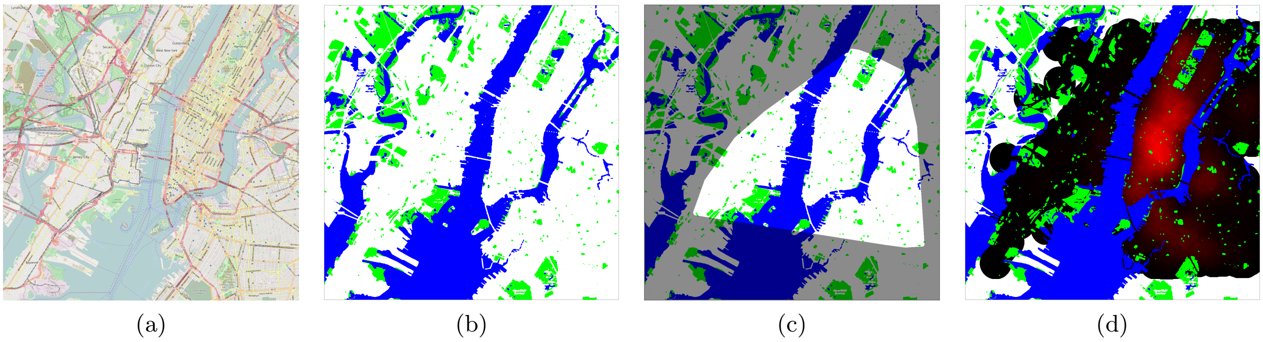

(a) (b) (c) (d)

Figure 9: Original and filtered versions of the OSM map of the target area in New York. The third figure from the left includes the convex hull spanned by 200m circles around the stations in the BSS — to improve the performance of the methods (particularly the regular grid and the Poisson and Ginibre point processes) we only draw stations inside the hull. The last figure includes the ratings, with red pixels indicating areas with a higher rating.

- $\max_t$: 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.

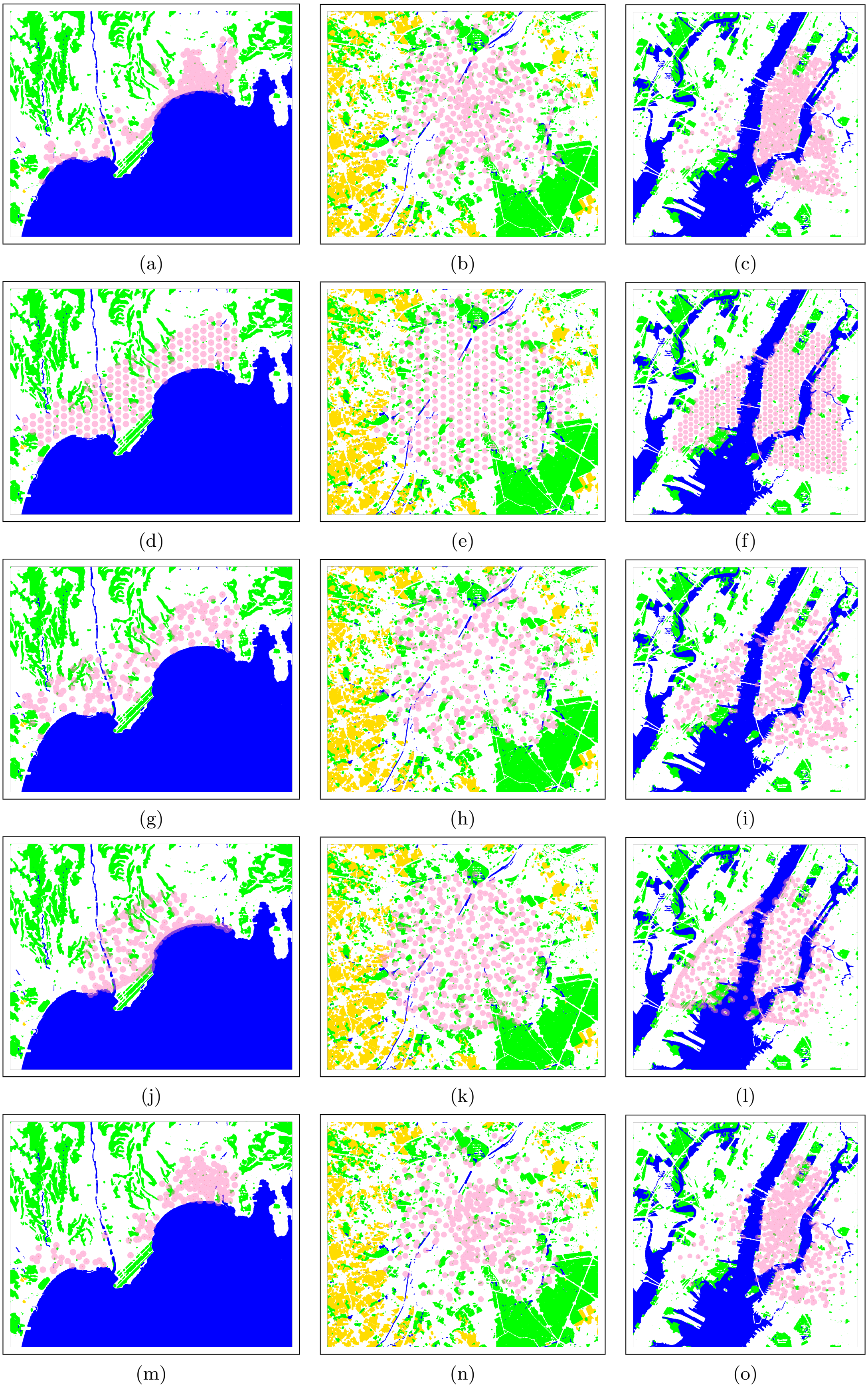
Table 2 presents these measures for a number of cities. We have made a selection from the close to 400 BSSs available on the `citybik.es` website, based on a number of basic metrics. First of all, the BSSs of *Barcelona*, *Paris*, *London*, and *New York* were considered because of their (large) size. 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 (see Figure 11b), as evidenced by its large value for $\max_t$, the largest edge length in the minimum spanning tree. In general, there is some degree of variety in terms of the inter-spacing, as evidence 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 10a to Figure 10b.

3.3 BSS topology generation

This procedure first requires determining the target area of the BSS, and as part of this, exclusion of areas that are not suitable for the placement of a station, such parks, bodies of water, forest and farmland. This procedure is done using the pixel-based tiles used to display OSM maps in a browser, and a convex hull is applied to the stations of the existing BSS to approximate its coverage. An example of this procedure applied to the BSS of New York is displayed in Figure 9. The algorithm to generate station locations assigns weights to pixels in the target area where a higher weight means a greater likelihood of selection as a station. A pixel is selected when its relative weight is greater than a random threshold, and the weights of each pixel change depend on the location of each new station. This is a very general procedure and can be applied to all three stochastic generation models. These include the Poisson point process [CI80], the Ginibre point process [Gin65] and the rating-weighted scheme which incorporates geographic information in the weights. Using a regular grid for placement is a deterministic approach that we also consider for comparison. Gaussian noise can be added to make the topology look less artificial in this case, although we have not done this in Figure 10 and Figure 11, which display results of the topology-generating procedures discussed above for six cities that have interesting characteristics, namely Nice, Brussels, New York, Dublin, Melbourne and Paris; and are a subset of those cities that appear in Tables 2 and 3.

3.4 Evaluation

In Table 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 10 and Figure 11). We also do that for the original BSS, and then check for how many pixels within the target area (the convex hull mentioned



(a) (b) (c)

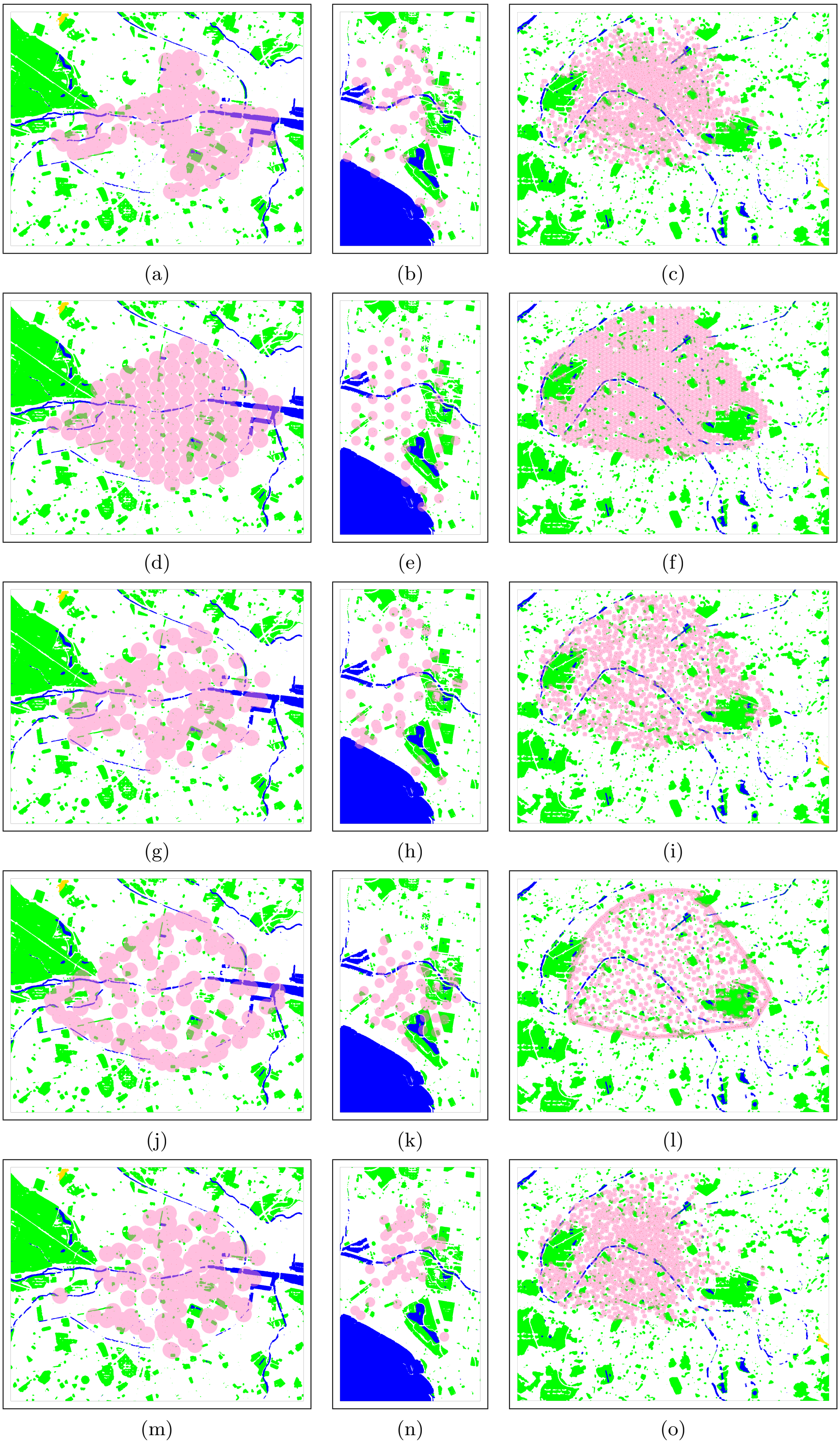
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 10: Left to right: Nice, Brussels, New York. Top to bottom: real system, regular grid (without noise), Poisson, Ginibre, rating-weighted.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 11: Areas covered by BSS topologies through 200m-radius circles around each station. Top to bottom: real system, regular grid (without noise), Poisson, Ginibre, rating-weighted. Left to right: Dublin, Melbourne, Paris.

City	Coverage Difference (200m)				Mean Absolute Difference			
	Grid	Poisson	Ginibre	Rated	Grid	Poisson	Ginibre	Rated
Barcelona	**.64**	.56 ± .01	.46 ± .01	**.64 ± .01**	**36.91**	41.77 ± 2.15	70.01 ± 0.18	37.56 ± 0.89
Brussels	.57	.60 ± .01	.60 ± .01	**.65 ± .01**	**31.05**	38.06 ± 1.86	37.13 ± 0.30	32.17 ± 0.93
Dublin	**.61**	.53 ± .02	.48 ± .01	.58 ± .01	**29.51**	38.57 ± 6.79	56.26 ± 0.66	35.43 ± 2.26
Glasgow	.64	.66 ± .01	.68 ± .01	**.71 ± .01**	26.50	32.32 ± 3.35	30.18 ± 1.91	**25.60 ± 1.59**
London	.57	.54 ± .01	.49 ± .01	**.60 ± .01**	**45.22**	49.94 ± 2.13	69.48 ± 0.30	47.79 ± 0.92
Melbourne	.69	.69 ± .01	.68 ± .01	**.73 ± .01**	**22.68**	27.60 ± 3.24	30.65 ± 0.84	23.41 ± 1.09
New York	.62	.62 ± .01	.56 ± .01	**.74 ± .01**	57.54	57.75 ± 1.50	76.30 ± 0.23	**44.61 ± 1.01**
Nice	.60	.62 ± .01	.63 ± .01	**.74 ± .01**	56.52	58.95 ± 2.29	82.82 ± 0.58	**49.63 ± 1.74**
Paris	.63	.57 ± .01	.50 ± .01	**.67 ± .01**	42.69	46.63 ± 1.55	63.04 ± 0.15	**41.20 ± 1.13**
Pisa	.73	.71 ± .01	.68 ± .01	**.74 ± .01**	**17.04**	25.46 ± 4.97	27.30 ± 3.42	21.15 ± 3.26
Tel Aviv	.59	.60 ± .01	.60 ± .01	**.65 ± .01**	43.40	50.48 ± 1.96	77.18 ± 0.40	**38.27 ± 1.27**

Table 3: Comparison of the topology generation methodologies in terms of the two metrics discussed in Section 3.4.

earlier) the original and generated BSS give the same result. This is divided by the total number of pixels in the convex hull to give an overlap score. We use a radius D of 200m in Table 3. 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 within 200 metres and a score of 1 to all pixels within 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). For both metrics, the numbers featured in Table 3 are the averages of 20 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, approximating a 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 Table 3.

For the 200m coverage overlaps, the rating-weighted scheme is nearly always the best choice, with Dublin the only exception. The regular grid does fairly well in all cases. For the mean absolute difference, the differences between the regular grid and the rating-weighted method are less pronounced. 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 Glasgow, Paris and Tel Aviv. It should be noted that there is still substantial room for parameter tuning for some methods (particularly Ginibre) — in fact, the Ginibre procedure used for the two metrics is slightly different (although this did not seem to impact the results to a noteworthy degree).

4 Using journey data for population discrete-space models

In some cases, one may have data that describes journeys made by agents and this can be used to establish parameters for a model. We consider two examples that have developed in the QUANTICOL project which illustrate two different approaches to using journey data.

4.1 Journey data for model reduction and parameterisation

In the QUANTICOL project, we have used real data to predict bike availability in BSSs [FHR16]. As mentioned in the introduction of this document, our initial approach to prediction has considered stations to behave as if isolated [GMRT15] which means that no spatial aspects of the data is taken into account. Currently, we are considering a different approach that does consider the relationship between stations spread over a region, and hence is spatial. Note that this is not specifically patch-based because it is not necessary to divide the region of interest into patches; it is sufficient to consider a bike station as the spatially located item of interest. The approach uses journey data to reduce the model size and thus to make analysis feasible. The approach uses journey data to reduce the model size and thus to make analysis feasible.

We consider London where there are around 780 bike stations. Analysis of a model which captures the full dynamics between all bike stations is computationally too expensive. Therefore, if we are interested in analysing the dynamics of a single target station at a time, we only need to model stations which make a significant contribution to the journey flows to that station. Ignoring the effect of a station which has a very small journey flow to the target station will have negligible impact on the accuracy of the model, and hence we can model these low contribution stations as a single integrated other station. We take historic journey data, and use this to assess the contribution of one station to the journey dynamics of another.

A directed weighted graph is constructed, in which each node represents a bike station and a weighted edge from node i to node j represents the direct contribution of station j to the journey flows to station i. Specifically, the weight of an edge from node i to node j is $w_{ij} = \lambda_i^j(t)/\lambda_i(t)$, where λ_i^j represents the bike arrival rate from station j to station i at time t; λ_i denotes the total bike arrival rate at station i at time t. Both λ_i^j and λ_i can be readily obtained from historical journey data. Furthermore, a contribution propagation method is applied to evaluate the indirect contribution of each bike station to another station. This (direct and indirect) contribution is compared to a threshold value, and edges are pruned from the graph if the contribution is too low. All stations that become disconnected in this process are then aggregated into a single 'other' bike station, thus reducing the model size. The final model then only captures the states of stations which have significant (direct or indirect) contribution to the journey flows to the target station. By doing this, we find that we only need to model around 20 stations to accurately analyse the dynamics of a target station.

Since bike stations have varying demands for acquisition or deposit of bikes at different times of day, it is necessary to split a day into slots of 20 minutes duration. We fit the following parameters with exponential distributions for each time slot.

- $\mu_i(k)$: The bike pickup rates at station i in time slot k.
- $\lambda_i(k)$: The bike arrival rates at station i in time slot k.
- $\lambda_i^j(k)$: The bike arrival rates at station i from station j in time slot k, $\lambda_i(k) = \sum_j \lambda_i^j(k)$.

In our model, the bike arrival rate at a station is reduced if another station which has significant direct journey flow to the station is empty. Preliminary results show that using this approach to model reduction provides better predictions that those obtained in the case where stations are considered in isolation.

4.2 Synthetic data for model transformation

Additionally, within the QUANTICOL project, we have considered how to take an individual-based continuous space model of information exchange between mobile nodes, and transform it to a population discrete-space model, to provide more scalable analysis. This transformation process requires that parameters be decided for the new model. ZebraNet is an ad hoc network where nodes are located in collars on wild animals and it has been modelled as individual animals moving in 2-dimensional space [JOW$^+$02, Fen12]. The model is interesting but is limited in terms of the number of nodes that can be analysed. It was possible to obtain journey data by simulating the movement of a few zebras (one per patch) to determine parameters to describe animal movement at the population level. The region of the model was not divided up into arbitrary squares but the location of waterholes were used to obtain a Voronoi partition (as illustrated in Figure 12) as the location of these affect the movement of the animals Thus parameters were estimated for the patch-based model, and this model was shown to have good correspondence with the original individual-based continuous space simulation [JOW$^+$02].

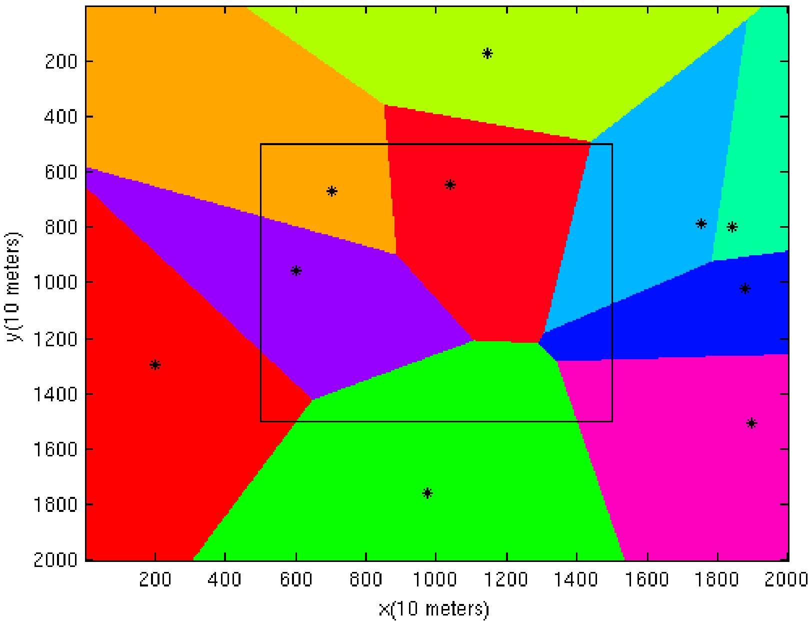


Figure 12: An example of a Voronoi tessellation of a landscape. Waterholes are indicated by asterisks and the rectangle indicates the route of the vehicle that gathers data from the animal collars.

The parameters were also used in a mean-field model derived from the patch-based model which was shown to be a good approximation. This research was influenced by a model of information spread between mobile nodes [CLBR09] which used real individual journey data and divided the region of interest into a grid of squares. This general type of approach can be seen as parameter extraction rather than parameter fitting, but it is a form of parameter estimation.

5 Conclusion

This deliverable has detailed the research in the QUANTICOL project where we have used spatial data to construct spatial models. Three distinct approaches have been considered and these involve GPS data to parameterise models of individual vehicle movement; static spatial analysis of location of stations in BSSs; and using journey data to obtain parameters for a model or to reduce the size of the model and the number of parameters that are to be estimated. This includes research that is complete (in the case of [Fen14]) and research that is almost complete [RG16, FHR16, Rei16] and has been or will shortly be submitted to appropriate venues for publication.

This deliverable relates to Task 2.3 which runs until the end of March 2016, and hence no further work specific to this task is envisioned in the project. However, the use of spatial data remains a feature of the project and the next section describes how this research may contribute directly or indirectly to other work packages.

5.1 Links with other work packages

Spatial data and related model construction and parameter estimation has obvious links with Work Package 5 which considers case studies, as well as tools. As mentioned earlier, some of the work relating to parameter estimation for routes has been embodied in tools produced in the QUANTICOL project [RGH15, RR15].

In terms of Work Package 4 which deals with Carma, the modelling language developed in the project, when we build models in Carma, then the use of spatial data for the construction of these models is important, and the work reported here contributes to this effort.

Work Package 3 considers spatial logics, and spatial data can be important for determining the formulae of interest that we wish our models to satisfy. Furthermore, the research into product families can be applied to the specification of BSSs, and the work presented in Section 3 could be combined with the product family research to extend the specification of new BSSs with spatial information.

Finally, Work Package 1 focusses on mean-field analysis techniques, as well as uncertainty. The techniques developed in Work Package 1 relating to uncertainty and imprecision in the parameters of Markov models can also be applied when we are considering parameters which are informed by spatial data.

A specific example of how this work on spatial data relates to a number of work packages, is ongoing research into a patch-based model of bus movement (WP5) expressed in Carma (WP4). This requires that parameters are obtained from real spatial data (WP2). The properties of this model can be evaluated using spatio-temporal logic for model checking (WP3). Other research in WP2 considers spatial analysis techniques, and these techniques could allow us to abstract away from certain details of the model so that it is possible to consider one ptach in detail, thus making the analysis scalable.

Acknowledgements

We thank Bill Johnston and Philip Lock of Lothian Buses, Edinburgh for providing access to the data and for their helpful feedback.

References (from the Quanticol project within the reporting period)

[Fen14] C. Feng. Patch-based hybrid modelling of spatially distributed systems by using stochastic HYPE – ZebraNet as an example. In *Proceedings of QAPL 2014*, 2014.

[FHR16] C. Feng, J. Hillston, and D. Reijsbergen. Moment-based probabilistic prediction of bike availability for bike-sharing systems, 2016. Submitted to a conference.

[GTV14] S. Gilmore, M. Tribastone, and A. Vandin. An analysis pathway for the quantitative evaluation of public transport systems. In *Proceedings of the 11th International Conference on Integrated Formal Methods (IFM 2014)*, LNCS 8739, pages 71–86. Springer, 2014.

[Rei16] D. Reijsbergen. Probabilistic modelling of station locations in bicycle-sharing systems, 2016. In preparation for conference submission.

[RG16] D. Reijsbergen and S. Gilmore. An automated methodology for analysing urban transportation systems using model checking, 2016. In preparation for journal submission.

[RR15] D. Reijsbergen and R. Ratan. Probabilistic modelling of the impact on bus punctuality of a speed limit proposal in Edinburgh. In *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2015)*, 2015.

References

[BDL04] G. Behrmann, A. David, and K.G. Larsen. A tutorial on UPPAAL. In *Formal methods for the design of real-time systems*, pages 200–236. Springer, 2004.

[BGME11] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 68–81. ACM, 2011.

[CI80] D.R. Cox and V. Isham. *Point processes*, volume 12. CRC Press, 1980.

[CLBR09] A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic. The age of gossip: spatial mean field regime. In *Eleventh International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance 2009)*, pages 109–120. ACM, 2009.

[CR14] E. Croci and D. Rossi. Optimizing the position of bike sharing stations. The Milan case. 2014.

[CZP$^+$15] L. Chen, D. Zhang, G. Pan, X. Ma, D. Yang, K. Kushlev, W. Zhang, and S. Li. Bike sharing station placement leveraging heterogeneous urban open data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 571–575. ACM, 2015.

[DLL$^+$11] A. David, K.G. Larsen, A. Legay, M. Mikučionis, and Z. Wang. Time for statistical model checking of real-time systems. In *Computer Aided Verification*, pages 349–355. Springer, 2011.

[DP73] D.H. Douglas and T.K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[EL14] C. Etienne and O. Latifa. 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):39, 2014.

[Fen12] C. Feng. *Modelling opportunistic networks with HYPE.* MSc dissertation, School of Informatics, University of Edinburgh, 2012.

[FGM12] C. Fricker, N. Gast, and A. Mohamed. Mean field analysis for inhomogeneous bike sharing systems. In *Aofa 2012, International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms*, 2012.

[Fis15] E. Fishman. Bikeshare: A review of recent literature. *Transport Reviews*, (ahead-of-print):1–22, 2015.

[FNO09] J. Froehlich, J. Neumann, and N. Oliver. Sensing and predicting the pulse of the city through shared bicycling. In *IJCAI*, volume 9, pages 1420–1426, 2009.

[FR15] I. Frade and A. Ribeiro. Bike-sharing stations: A maximal covering location approach. *Transportation Research Part A: Policy and Practice*, 82:216–227, 2015.

[Gal15] V. Galpin. Quantitative modelling of residential smart grids. In *Software Engineering and Formal Methods - SEFM 2015 Collocated Workshops: ATSE, HOFM, MoKMaSD, and VERY*SCART, Revised Selected Papers*, LNCS 9509, pages 161–175. Springer, 2015.

[Gin65] J. Ginibre. Statistical ensembles of complex, quaternion, and real matrices. *Journal of Mathematical Physics*, 6(3):440–449, 1965.

[GMRT15] N. Gast, G. Massonnet, D. Reijsbergen, and M. Tribastone. Probabilistic forecasts of bike-sharing systems for journey planning. In *Proceeding of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15)*. ACM, 2015.

[GPGL12] J.C. García-Palomares, J. Gutiérrez, and M. Latorre. Optimizing the location of stations in bike-sharing programs: a GIS approach. *Applied Geography*, 35(1):235–246, 2012.

[GR15] S. Gilmore and D. Reijsbergen. Validation of automatic vehicle location data in public transport systems. *Electronic Notes in Theoretical Computer Science*, 318:31–51, 2015.

[Jen67] George F. Jenks. The data model concept in statistical mapping. *International yearbook of cartography*, 7(1):186–190, 1967.

[JOW$^+$02] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. *SIGPLAN Notices*, 37:96–107, 2002.

[JTS13] S. Jäppinen, T. Toivonen, and M. Salonen. Modelling the potential effect of shared bicycles on public transport travel times in Greater Helsinki: An open data approach. *Applied Geography*, 43:13–24, 2013.

[LAC12] N. Lathia, S. Ahmed, and L. Capra. Measuring the impact of opening the London shared bicycle scheme to casual users. *Transportation research part C: emerging technologies*, 22:88–102, 2012.

[LCG15] L. Luisa Vissat, A. Clark, and S. Gilmore. Finding optimal timetables for Edinburgh bus routes. *Electronic Notes in Theoretical Computer Science*, 310:179–199, 2015.

[LLQ+] J. Liu, Q. Li, M. Qu, W. Chen, J. Yang, X. Hui, H. Zhong, and Y. Fu. Station site optimization in bike sharing systems.

[MD] R. Meddin and P. DeMaio. The bike-sharing world map. `https://www.google.com/maps/d/viewer?mid=zGPlSU9zZvZw.kmqv_ul1MfkI`. Accessed: 2015-01-28.

[OCB14] O. O'Brien, J. Cheshire, and M. Batty. Mining bicycle sharing data for generating insights into sustainable transport systems. *Journal of Transport Geography*, 34:262–273, 2014.

[Pli92] J.L. Pline. *Traffic engineering handbook*. Prentice-Hall, 1992.

[R D05] R Development Core Team. R: A language and environment for statistical computing. 2005.

[RG14] D. Reijsbergen and S. Gilmore. Formal punctuality analysis of frequent bus services using headway data. In *Proceedings of the 11th European Workshop on Computer Performance Engineering (EPEW 2014)*, LNCS 8721, pages 164–178. Springer, 2014.

[RGH15] D. Reijsbergen, S. Gilmore, and J. Hillston. Patch-based modelling of city-centre bus movement with phase-type distributions. *Electronic Notes in Theoretical Computer Science*, 310:157–177, 2015.

[RKW12] P. Reinecke, T. Krauß, and K. Wolter. HyperStar: Phase-type fitting made easy. In *Proceedings of the Ninth International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 201–202, 2012.

[Sg09] The Scottish government. Bus Punctuality Improvement Partnerships (BPIPs) guidance document. 2009.

[ZZB+13] Y. Zhang, M. Zuidgeest, M. Brussel, R. Sliuzas, and M. van Maarseveen. Spatial location-allocation modeling of bike sharing systems: A literature search. In *Proceedings of the 13th World Conference on Transportation Research*, 2013.