

Placement of 5G Drone Base Stations by Data Field Clustering

Stefano Iellamo
ICS-FORTH
Leof. Plastira 100, Greece
siellamo@ics.forth.gr

Janne J. Lehtomäki
University of Oulu
Centre for Wireless Communications, Finland
jannel@ee.oulu.fi

Zaheer Khan
University of Liverpool,
EEE Department, UK
zaheer.khan@liverpool.ac.uk

Abstract—We consider the problem of complementing the capacity of an existing network of macro base stations by dynamically placing a network of 5G small base stations in the form of Unmanned Aerial Vehicles UAV (better known as drones). Our goal is to maximize the capacity boost provided by the UAVs in each considered time frame and extend the battery life of the served mobile users. With this in mind, we propose two clustering algorithms that build on mobile users' spatio-temporal data excess demand (here intended as the portion of demand which is not satisfactory addressed by the existing macro base stations). For the numerical analysis, we use real Beijing downtown trajectory data. The obtained results show that our algorithms perform well and can be considered for enabling real time connection provisioning.

I. INTRODUCTION

Better coverage and capacity requirements of future cellular networks may not be fully satisfied by traditional fixed terrestrial base station (BS) deployments. Moreover, deployment of a very dense network of BSs to provide high capacity and coverage is not suitable in terms of CAPEX and OPEX, due to the fact that: 1) user wants better service at affordable price; and 2) a high percentage of densely deployed BSs will be lightly loaded most of the time and space. To address such challenges, the utilization of Unmanned Aerial Vehicles or drone BSs (here on referred to as UAV-BSs) is a promising solution. One of the greatest challenges involved in the UAV-BS deployment is to find the optimal placement for the UAV-BSs at each time instant by relying on mobile users' spatial demand.

Obtaining and using mobile users' spatial demand is becoming more and more important in wireless communications. Such information can in fact be used by the operators for a variety of purposes, such as deployment and design of new network elements (e.g., 5G small cells, relays and Internet of Things Gateways) for punctual coverage, network resource allocation, indoor coverage planning and better interference management. However, obtaining punctual information on the users' radio conditions has historically been a challenging task. Nowadays, the availability of GPS in the vast majority of user terminals (UT) allows retrieving this information when it is available. Thanks to the 3GPP functionality "Minimization of Drive Tests (MDT)" [1] the

UTs can, upon network request, send a series of radio quality indicators (coverage, quality of service as a function of the service etc.) by associating them to their geo-location. The first field tests of the MDT function have started in 2016 and many operators are envisioning a commercial deployment by 2018-2019.

In this paper, we leverage mobile users' spatial *excess* demand (i.e., the demand that is not satisfied by the macro base stations, here on simply referred to as demand) to design algorithms for dynamically placing 5G UAV-BSs. Our goal is to capture the spatio-temporal relation between data demand points and depict the clusters (here intended as an agglomerate of demand points served by a UAV-BS) and their centers during well-chosen time frames.

Generally speaking, clustering algorithms attempt to classify elements into categories (the clusters), on the basis of their similarity. Several different clustering strategies have been proposed in the literature, though the definition of cluster may vary. In K-means [2] and K-medoids [3] methods, for instance, clusters are groups of data characterized by a small distance to the cluster center. An objective function, typically the sum of the distance to a set of putative cluster centers, is optimized until the best cluster centers candidates are found. However, because a data point is always assigned to the nearest center, these approaches are not able to detect nonspherical clusters [4]. In distribution-based algorithms, one attempts to reproduce the observed realization of data points as a mix of predefined probability distribution functions [5]; the accuracy of such methods depends on the capability of the trial probability to represent the data. Clusters with an arbitrary shape are easily detected by approaches based on the local density of data points. In density-based spatial clustering of applications with noise (DBSCAN) [6], one chooses a density threshold, discards as noise the points in regions with densities lower than this threshold, and assigns to different clusters disconnected regions of high density. However, choosing an appropriate threshold can be non-trivial, a drawback not present in the mean-shift clustering method [7], [8]. There a cluster is defined as a set of points that converge to the same local maximum of the density distribution function. This method allows the finding of nonspherical clusters but works only for data defined by a

set of coordinates and is computationally costly.

In this paper, we propose an alternative approach by drawing inspiration from [9], which is in turn based on quick density peak search [10] and on the theory of data field [11]. The advantages brought by our algorithms are in that they 1) automatically determine cluster parameters, 2) can be effectively applied to work with irregular spatial distribution of demand data points (such as trajectory data), and 3) they have very low computational complexity. Besides, by associating input data to a specific time span, they could allow to track in real time the evolution of the clusters and their respective centers, this by relying on the instantaneous spatial demand. Such information could be readily used by mobile operators to boost the capacity of their existing networks by designing efficient and effective UAV-BS migration strategies.

With this paper, our contribution is two-fold:

- By drawing inspiration from an algorithm recently proposed to solve the taxi hotspot problem, we design two clustering algorithms for the real-time placement of 5G UAV-BSs which are complementary to already existing macro base stations. To the best of our knowledge, we are the first using data field clustering for dynamic resource (in our particular case, UAV-BSs) allocation in the context of wireless cellular networks.
- We simulate the behavior of our algorithms relying on real people's trajectories in Beijing downtown.

II. MODEL AND PROBLEM FORMULATION

We model an operator's served region as a 2D space where mobile users move while demanding data service. We assume that demand points can be characterized by means of data mining techniques utilizing, e.g., GPS traces, cell load trends and handovers¹.

In this paper, we propose an algorithm that can work by taking as input either trajectory demand points (atomic level) or statistical demand points (aggregate level).

Definition 1 (Trajectory demand point). *A mobile user's trajectory demand point is defined as a geographical point which is part of a mobile user's trajectory and is representative of that mobile user's spatial demand only in a given time span.*

Definition 2 (Statistical demand point). *A statistical demand point is defined as a geographical point which is associated to the aggregate demand of all the mobile users passing by that point in a given time span and requesting data services.*

Let us denote by $\mathcal{P}(t)$ a data set made of n data demand points $\{P_1(t), P_2(t), \dots, P_n(t)\}$ related to time period t . Note that in order to lighten notation we will some time omit the iteration index if the latter refers to the current iteration t . Each point is modeled as a particle characterized by a mass and a virtual field around it. The potential of each point depends on

the interaction measure of that point with the points located in its vicinity. The potential value ϕ of the demand point P_i can be expressed as in [12]:

$$\phi(P_i) = \sum_{j=1}^n m_j \cdot e^{-\left(\frac{d_{ij}}{\sigma}\right)^k} \quad (1)$$

where m_j is the mass of the demand point P_j , d_{ij} is the distance between P_i and P_j , $k \in \mathbb{N}$ is the distance index, and $\sigma > 0$ is the impact factor. The latter controls the distance of interaction between data objects and has a strong impact on the spatial distribution of data field. Please note that the mass must satisfy $m_i \geq 0 \forall i$, $\sum_i^n m_i = 1$.

It has been proven [12], [11] that spatial distribution of data field is insensitive to the value of k , and strongly depends on σ . Thus, by setting $k = 2$ we get a Gaussian potential function which has desirable mathematical properties.

In our case, we model each demand point i as a particle characterized by a certain mass (demand in bits normalized by the sum of all demands²) and a certain potential during a given time span. Please recall that in the trajectory clustering case the demand is wrt a single user, while in the case of statistical demand points is wrt multiple mobile users.

III. THE CLUSTERING MECHANISM

We adapt a method of trajectory clustering based on decision graphs and data fields which was recently proposed to solve the taxi hotspot problem. This method is particularly suitable in our case because it takes as input uneven demand points (in [9] these are the moving users' trajectories) and assigns to each point (or to a subset of points) a potential, which is a function of the local data demand point density and mass. A drawback of [9] is that it doesn't take into consideration the spatio-temporal relation of the trajectory points. To cope with this problem, one can draw trajectories corresponding to well-chosen time periods, run the trajectory clustering algorithm for each period and then put the results one after the other (like for time-lapse photography).

The algorithm proposed in [9] builds on [10] and [12], which were the first to apply field theory to users' interaction data mining (the former) and to propose a clustering method relying on decision graphs (the latter). In this paper we adapt the algorithm proposed in [9] to find the optimal position for UAV-BSs.

A. Impact factor optimization

We would like the impact factor to guarantee the lowest uncertainty wrt clusters formation; in other words, we want to minimize the entropy. The latter can be rewritten from Shannon [13] as follows:

$$H = - \sum_{i=1}^n \frac{\phi(P_i)}{Z} \log \left(\frac{\phi(P_i)}{Z} \right) \quad (2)$$

¹The way info on user data demand patterns are collected is out of the scope of this paper.

²If the algorithm is run iteratively the sum of demands must be updated so as to satisfy the constraint $\sum_i^n m_i = 1$

where $\phi(P_1), \phi(P_2), \dots, \phi(P_n)$ are the potential values of the data objects P_1, P_2, \dots, P_n , $Z = \sum_{i=1}^n \phi(P_i)$ and $0 \leq H \leq \log(n)$.

From Eq. (2) one can notice that the maximum entropy is reached when $H = \log(n)$ and $\phi(P_1) = \phi(P_2) = \dots = \phi(P_n)$ and that the minimum entropy corresponds to the maximum possible heterogeneity among potential values. Thus, if the potential values are asymmetric, the system experiences low uncertainty, i.e., the entropy level is low.

It is shown in [11] that $H(\sigma)$ is a concave function. Therefore, several search methods (e.g., numerical and heuristic methods) can be used to find the optimal σ^* that minimize the entropy. However, in practical implementations the use of the optimal σ^* may expose the clustering process to noise and outlier data. For this reason, the final σ^{\otimes} is often chosen to be slightly larger than its optimal value, i.e., $\sigma^{\otimes} = c \cdot \sigma^*$, ($1 < c \leq 2$).

B. Clusters and centers selection by decision graphs

We define cluster centers as local density maxima (i.e., data objects with relatively higher potential value) that are 'far' from demand points characterized by higher density [10]. In this way each center depends only on the points relative densities rather than on their absolute value. Thus, let us define ρ_i as the local density of point i .

$$\rho_i = \sum_{j \in \mathcal{P}} \chi(d_{ij} - d_c) \quad (3)$$

where d_c is the cutoff distance, d_{ij} is the distance between demand point i and point j , and $\chi(x)$ is a function which is equal to 1 if $x < 0$, and equal to 0 otherwise (i.e., for $x \geq 0$). Basically, ρ_i is equal to the number of points that are closer than d_c to point i . The algorithm is sensitive only to the relative magnitude of ρ_i in different points, implying that, for large data sets, the results of the analysis are robust with respect to the choice of d_c ³.

Let δ_i denote the distance between point i and the nearest point v with higher density.

$$\delta_i = \min_{v: \rho_v > \rho_i} d_{iv} \quad (4)$$

For the global maximum (i.e., the point characterized by the highest density), we set ρ_i to a very large number (e.g., $\rho_i = \max_v d_{iv}$).

IV. THE ALGORITHMS

A. The max-capacity clustering algorithm

The first clustering algorithm we propose is able to spot clusters of any size and shape and therefore maximizes the capacity boost the mobile network operator can achieve if such clusters are duly served. That means, a coverage optimization problem will have to be set up and solved following the

Algorithm 1 Max-capacity clustering algorithm for 5G UAV-BS placement

- 1: **Initialization:** Set the duration t of a time period, the masses $m_i(t)$ for all t , the cutoff distance d_c .
- 2: **while** At each time period t **do**
- 3: Obtain the optimal impact factor σ^* that minimizes the entropy potential $H(\sigma)$, and compute the potential value for each data demand point according to Eq. (1).
- 4: Compute $\delta_i(t)$ for each demand point i (Eq. (4)).
- 5: Obtain the threshold values δ_{th} , ρ_{th} , ϕ_{th}^{sup} and ϕ_{th}^{inf} .
- 6: Identify the clusters centers, i.e., all those points i characterized by $\phi_i(t) > \phi_{th}^{sup}(t)$ and $\delta_i(t) \geq \delta_{th}$.
- 7: Identify and clear the noise points, i.e., all points i characterized by $\phi_i(t) < \phi_{th}^{inf}(t)$.
- 8: Form the clusters: Assign each remained data object to the nearest centre characterized by higher potential value.
- 9: Cover the formed clusters by duly placing and operating a set of UAV-BSs.
- 10: **end while**

clusterization algorithm that we present here. The main steps of the max-capacity clustering algorithm (Algorithm 1) wrt a time frame t are detailed here below.

1. **Obtain the optimal impact factor σ^* that minimizes the entropy potential $H(\sigma)$.** This can be done, e.g., by means of the search method proposed in [11].
2. Using the optimal impact factor σ^* , compute the potential value for each data demand point according to Eq. (1).
3. **Compute density values and threshold values for each demand point i .** Threshold values δ_{th} , ρ_{th} , ϕ_{th}^{sup} and ϕ_{th}^{inf} can be obtained by means of the modified elbow method [14].
4. **Find the cluster centers.** These are the points i with local maximum potential value (for convenience it is possible to use a threshold method so that one can label as local potential maxima all those points i characterized $\phi_i(t) > \phi_{th}^{sup}(t)$ and $\delta_i(t) \geq \delta_{th}$).
5. **Identify and clear the noise points.** Since noise points usually scatter in data field and receive weak mutual interaction, they have lower potential values. Thus, clear all points i characterized by $\phi_i(t) < \phi_{th}^{inf}(t)$.
6. **Form the clusters:** Assign each remained data object to the nearest centre characterized by higher potential value.
7. Cover the formed clusters by duly placing a set of UAV-BSs.

A visual example of the decision graph phase, cluster formation phase and drone covering optimization phase is shown in Fig. 1 (toy dataset retrieved from [15]).

B. The iterative clustering algorithm

Ideally, we would like the algorithm to tell us where exactly the UAV-BSs should be placed, rather than the areas to be served so as maximize the capacity boost. Hence, we propose an iterative version of the max-capacity algorithm which:

- generates clusters whose extension from the center is proportional to the UAV-BS maximum coverage, and

³Scenario-dependent techniques for fixing d_c in can be found in [10].

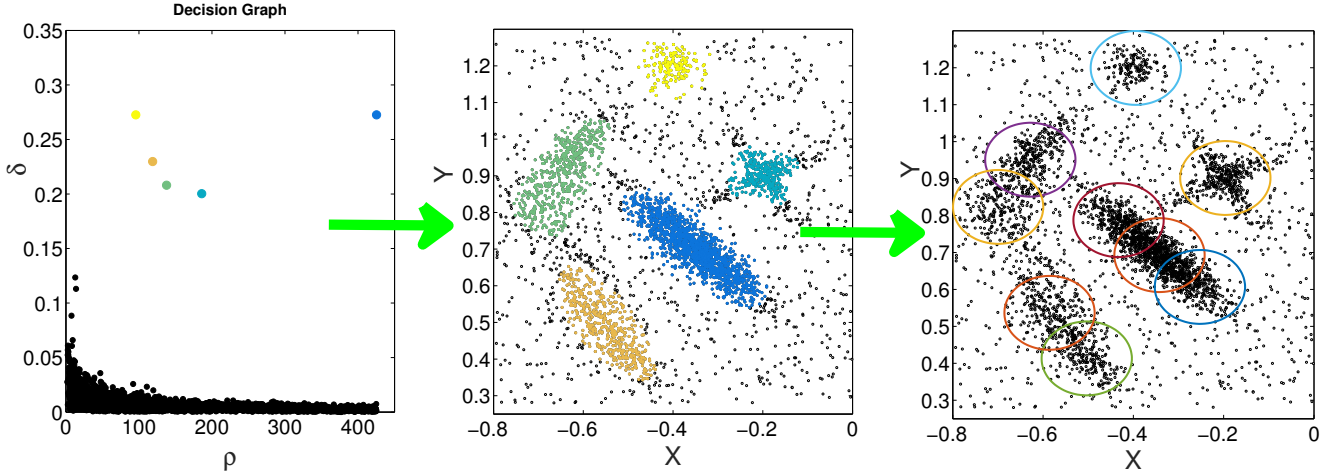


Fig. 1. Visual example of Algorithm 1 phases: decision graph phase (left), cluster formation phase (center) and drone covering optimization phase (right). Dataset is from [15].

Algorithm 2 Iterative clustering algorithm for 5G UAV-BS placement

- 1: **Initialization:** Set the duration t of a time period, the masses $m_i(t)$ for all t and for all i , the minimum distance d_{min} between two BSs and the number N_u of available UAV-BSs. Set the cutoff distance d_c equal to the maximum UAV-BS coverage distance.
 - 2: **while** At each time period t **do**
 - 3: **while** There is still excess demand AND there are still available UAV-BS **do**
 - 4: Obtain the optimal impact factor σ^* that minimizes the entropy potential $H(\sigma)$, and compute the potential value for each data demand point according to Eq. (1).
 - 5: Choose the threshold δ_{th} as the first elbow point greater than d_{min} , i.e., $\delta_{th} = \min_{\delta_{elbow} > d_{min}} \{\delta_{elbow}\}$
 - 6: Obtain ρ_{th} and ϕ_{th}^{sup} with the modified elbow method [14].
 - 7: Identify the clusters centers. These are the points i characterized by $\phi_i > \phi_{th}^{sup}$ and $\delta_i > \delta_{th}$.
 - 8: Assign each demand point to the nearest issued cluster center within its coverage range.
 - 9: Order the issued cluster centers by decreasing density δ_i and assign a UAV-BS to the centers characterizes by higher densities first.
 - 10: Clear all demand points that have been assigned to a cluster center (i.e., that are served by a UAV-BS) and update the number of available UAV-BSs.
 - 11: **end while**
 - 12: **end while**
-

whose center is indicative of the point where the UAV-BS should be placed

- takes into account the number of available UAV-BSs.

The iterative clustering algorithm is detailed in Algorithm 2

V. SIMULATION ANALYSIS

For the simulation analysis, we would have liked to test the performance of our algorithm on real MDT data related to many users over consecutive limited time spans. However, the mobile operators we have contacted have either not replied or not granted us the use of their data sets.

Thus, in order to provide an idea of the functionality of our algorithms, we fall back on the GeoLife dataset [16] which includes the GPS trajectories of 182 users over five years (from April 2007 to August 2012). A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude and altitude. Although such data are of course very far from being a perfect fit to test our algorithms (they are representative of few users over a very long period, and there is no timely mass-user correspondence), we believe this can be taken as a first step to show the potential of our algorithms. We hope to be able, in the near future, to build a simulator running with real time MDT data.

Looking at the results achieved by the two algorithms (Fig. 2-left for Algorithm 1 and Fig. 2-right for Algorithm 2), we notice right away the difference in terms of coverage. In fact, despite we have allocated 30 UAV-BSs for Algorithm 2, the latter is able to cover nearly 46% of the total mass, against a larger nearly 62% that would be reached by covering the clusters spotted by Algorithm 1 with a set of drones. This because the iterative algorithm, if on the one hand does not require coverage optimization for clusters covering (the cluster centers are the recommended positions for UAV-BS placement), on the other hand may end up in minor capacity boosts, the number or utilized UAVs being equal. This because it tends to generate clusters whose size is proportional to the UAV coverage range and which are

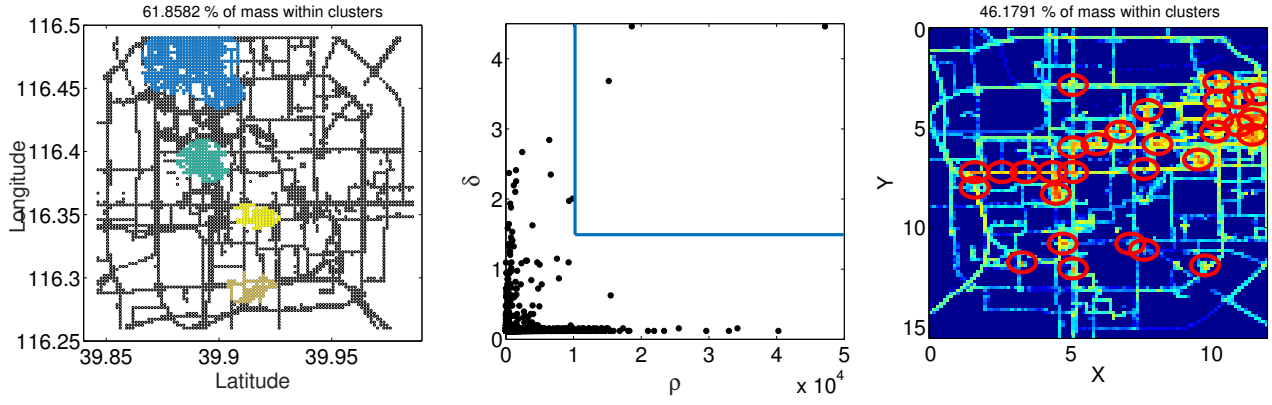


Fig. 2. Algorithm 1 (left) and Algorithm 2 (right) performance with the GeoLife dataset as input. Masses are chosen equal to one for all trajectory points. Settings for the iterative algorithm: $N_u = 30$, $d_c = 500m$, $d_{min} = 1Km$. The center figure is the first iteration decision graph which contribute (along with other 5 iterations) to generate the figure on the right hand side.

geometrically centered at points of local potential maximum. In the case of city scenarios and trajectory data inputs, centers will be very often the crossing between main streets, as one can infer from Fig. 2-right.

Fig. 2-center show the decision graph of the first iteration (out of five) which has led to Fig. 2-right (Algorithm 2). The automatic threshold selection, which occurs at each iteration, is for this iteration evidenced by a blue frame.

VI. CONCLUSION AND FUTURE WORK

We have considered the problem of complementing the capacity of an existing network of macro base stations by dynamically placing Unnamed Aerial Vehicles Base Stations. We have proposed two data field clustering algorithms that build on mobile users' spatio-temporal data excess demand, i.e., that portion of demand which is not satisfactory addressed by the existing macro base stations. For the numerical analysis, we have tested our algorithms with Beijing downtown mobility patterns. We have shown that the two algorithms require very low computational capabilities and perform very well though differ in some critical points. Furthermore, they require low computational complexity, and are therefore particularly suitable for operations involving large datasets. As a future research direction, we plan to build a demo relying on real 3GPP Minimization of Drive Tests Data and to integrate the presented algorithms in the context of large Internet of Things (IoT) networks. In fact, IoT nodes would greatly benefit from the optimization of the IoT Gateway position, both in terms of battery life (intuitively, less power is needed if the IoT Gateway is closer) and scalability (e.g., in the case of dense IoT clusters).

REFERENCES

- [1] W. A. Hapsari, A. Umesh, M. Iwamura, M. Tomala, B. Gyula, and B. Sebire. Minimization of drive tests solution in 3GPP. *IEEE Communications Magazine*, 50(6):28–36, June 2012.
- [2] J. MacQueen. Some methods for classification and analysis of multivariate observations. The Regents of the University of California, 1967.

- [3] Leonard Kaufman and Peter Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, March 1990.
- [4] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, June 2010.
- [5] J. McLachlan and T. Krishnan. *The EM algorithm and Extension*. Wiley inter-science, 1996.
- [6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.
- [7] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, January 1975.
- [8] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, August 1995.
- [9] P. X. Zhao, K. Qin, Q. Zhou, C. K. Liu, and Y. X. Chen. DETECTING HOTSPOTS FROM TAXI TRAJECTORY DATA USING SPATIAL CLUSTER ANALYSIS. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-4/W2:131–135, July 2015.
- [10] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, June 2014.
- [11] Deren Li, Shuliang Wang, Wenyan Gan, and Deyi Li. Data Field for Hierarchical Clustering. *Int. J. Data Warehous. Min.*, 7(4):43–63, October 2011.
- [12] Deyi Li and Yi Du. *Artificial Intelligence with Uncertainty*. 1 edition.
- [13] C. E. Shannon. A Mathematical Theory of Communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001.
- [14] Yihong Yuan and Martin Raubal. Measuring Similarity of Mobile Phone User Trajectories— a Spatio-temporal Edit Distance Method. *Int. J. Geogr. Inf. Sci.*, 28(3):496–520, March 2014.
- [15] Alessandro Laio. http://people.sissa.it/laio/Research/Res_clustering.php.
- [16] Xing Xie Wei-Ying Ma Yu Zheng. GeoLife: A Collaborative Social Networking Service among User, location and trajectory. *IEEE Data(base) Engineering Bulletin*, June 2010.