Who Cares about Others' Privacy: Personalized Anonymization of Moving Object Trajectories

Despina Kopanaki
Dept. of Informatics
University of Piraeus, Greece
dkopanak@unipi.gr

Vasilis Theodossopoulos
Dept. of Informatics
University of Piraeus, Greece
bill.theodossopoulos@gmail.com

Nikos Pelekis
Dept. of Statistics and Insurance Sc.
University of Piraeus, Greece
npelekis@unipi.gr

Ioannis Kopanakis
Dept. of Business Administration
Tech. Educational Institute of Crete, Greece
i.kopanakis@teicrete.gr

Yannis Theodoridis
Dept. of Informatics
University of Piraeus, Greece
ytheod@unipi.gr

ABSTRACT

The preservation of privacy when publishing spatiotemporal traces of mobile humans is a field that is receiving growing attention. However, while more and more services offer personalized privacy options to their users, few trajectory anonymization algorithms are able to handle personalization effectively, without incurring unnecessary information distortion. In this paper, we study the problem of *Personalized (K,Δ)-anonymity*, which builds upon the model of (*k*,*δ*)-anonymity, while allowing users to have their own individual privacy and service quality requirements. First, we propose efficient modifications to state-of-the-art *(k,δ)*-anonymization algorithms by introducing a novel technique built upon users' personalized privacy settings. This way, we avoid over-anonymization and we decrease information distortion. In addition, we utilize dataset-aware trajectory segmentation in order to further reduce information distortion. We also study the novel problem of *Bounded Personalized* (*K*,*Δ*)-*anonymity*, where the algorithm gets as input an upper bound the information distortion being accepted, and introduce a solution to this problem by editing the *(k,δ)* requirements of the highest demanding trajectories. Our extensive experimental study over real life trajectories shows the effectiveness of the proposed techniques.

Keywords

Moving objects databases; Trajectories; *k*-anonymity; Personalization; Uncertainty; Segmentation; Distortion.

1. INTRODUCTION

With the rapid development of information and communication technologies, the advent of mobile computing and the increasing popularity of location-aware services, the volume of mobility data gathered daily by service providers has exploded. It is safe to predict that this trend will continue in the near future. Publishing such information allows researchers to analyze humans' trajectories and extract behavioral patterns from them, in order to support decision-making.

However, publishing datasets consisting of humans' trajectories creates threats regarding the privacy of the individuals involved. This occurs when the spatiotemporal traces that users leave behind are combined with other publicly available information, which can reveal their identity, as well as other sensitive information about them (place of residence, sexual orientation, religious or political beliefs, etc.). Thus, it becomes necessary to develop methods providing privacy-preservation in mobility data publishing, where a sanitized version of the original dataset is published while the maximum possible data utility is maintained. A number of anonymization methods have been proposed so far, with most of them adopting the concept of *k-anonymity*, the fundamental principle which states that every entry of a published database should be indistinguishable from at least *k*–1 other entries. For example, trajectories are grouped into clusters of at least *k* members and published as cylindrical volumes which 'conceal' the individual trajectories [1][2], points of trajectories are suppressed so that adversaries with partial knowledge of a trajectory cannot identify a specific one amongst at least *k*–1 others [13], and so on.

Figure 1(a) illustrates an example dataset consisting of 5 trajectories, where each trajectory is associated with its own *k*-anonymity requirement, whereas Figure 1(b) illustrates the anonymization provided by W4M [2], a state-of-the-art (*k*,*δ*)-anonymity algorithm, assuming (a universal) *k* = 3 requirement (i.e., the maximum of the particular requirements). Clearly, the result fails to maintain the trend of the original data. However, if we could have taken into account the specific users' privacy preferences (i.e., the different *k*'s in Figure 1(a)), two clusters instead of one would have been created, as illustrated in Figure 1(c); *this is the first objective of this paper*. Moreover, if we could have performed an appropriate segmentation of trajectories in sub-trajectories before the anonymization process, the distortion would be even less, as illustrated in Figure 1(d); *this is the second objective of this paper*. (In this example, the *δ*- parameter effect is not discussed but it is similar to that of *k*.)

Series ISSN: 2367-2005

10.5441/002/edbt.2016.39

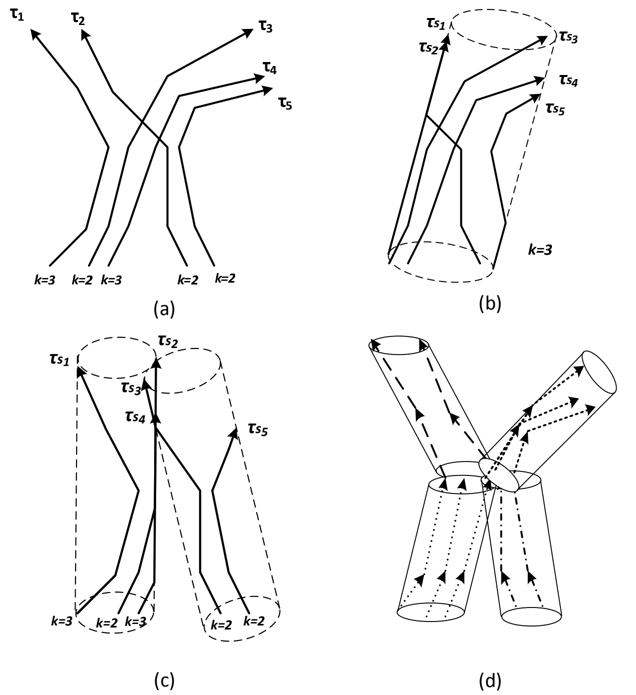


Figure 1: (a) a set of five trajectories along with the anonymization result provided by (b) universal k; (c) personalized k_i; (d) segmentation and personalized k_i.

As revealed by the previous example, an important drawback of most of the existing anonymization methods is that they are based on universal (e.g. k = 3) rather than user-defined privacy requirements. This lack of personalization may lead to unnecessary anonymization and data utility loss for users whose privacy requirements are overvalued and to inadequate anonymization and violation of privacy for users whose requirements are undervalued. Towards this goal, state-of-the-art techniques, such as [1][2] can be extended to use a user-specific privacy threshold k and uncertainty diameter δ. On the other hand [9] offers personalization by introducing trajectory-specific privacy requirements, however it ignores service quality, as it will be discussed in Section 2. In contrast, the method we propose uses trajectory-specific values to determine each user's specific privacy level and service quality requirements, therefore reducing data utility loss and improving service quality.

An additional shortcoming of the existing anonymization methods that are based on clustering is that they function at the trajectory level. As a result, when dealing with trajectories that are on the whole very different to each other, though they maintain some similar parts, these algorithms fail to recognize this situation and either assign such trajectories to different clusters (i.e., k-anonymity sets) or assign them into the same cluster after considerable spatiotemporal translation. This failure to recognize and make use of similarities between parts of trajectories is counter-intuitive and increases the overall distortion. In our proposal, we deal with this problem by utilizing trajectory segmentation in order to discover similar sub-trajectories and use those as the basis of our clustering process.

In this paper, we present the so-called *Who-Cares-about-Others'-Privacy* (WCOP) suite of methods for publishing spatiotemporal trajectory data using personalized (K,Δ)-anonymity, extending the concept of (k,δ)-anonymity as introduced in [1][2], where, for each user u_i, k_i dictates the required privacy level of the specific user and δ_i functions as a service quality threshold. In addition, we adopt a privacy-aware trajectory segmentation phase, during which trajectories are partitioned into sub-trajectories. This phase allows the clustering algorithm to discover similarities between the trajectories and assign the respective partitions into clusters, the members of which require the least necessary editing to fulfill (K,Δ)-anonymity, thus keeping distortion as low as possible. Finally, we present an approach aiming at controlling the information loss caused by the anonymization. The most *demanding* trajectories, i.e., the ones corresponding to users who require to be hidden among a large number of other users (hence, high k) in a small region (thus, low δ), are edited in order to be made less demanding, thus decreasing anonymization distortion.

Summarizing, in this paper, we make the following contributions:

- We propose WCOP-CT, an algorithm that extends [1][2] for spatiotemporal trajectory data publication based on the assumption that users have different privacy preferences.
- We extend WCOP-CT to WCOP-SA, by incorporating a trajectory segmentation phase aiming to facilitate the discovery of patterns shared between parts of trajectories, thus decreasing the distortion caused during the anonymization process.
- We propose WCOP-B, an algorithm able to control the level of the anonymization distortion by data assessment and requirements relaxation.
- Finally, we conduct a comprehensive set of experiments over a real trajectory dataset, in order to evaluate our approach.

The rest of the paper is structured as follows: Section 2 presents related work. Section 3 formulates the two problems to be addressed: *Personalized (K,Δ)-anonymity* and *Bounded Personalized (K,Δ)-anonymity*, respectively. Sections 4 and 5 provide effective solutions to the above two problems, respectively. Our experimental study is presented in Section 6. Finally, Section 7 concludes the paper.

2. RELATED WORK

The methods proposed so far in order to tackle the issue of privacy-preserving mobility data publishing mostly adopt the principle of k-anonymity, which was originally proposed for relational databases [12]. In the context of mobility data, trajectories of moving objects are time ordered sequences of (p, t) pairs, where p denotes the place a moving object was located at recorded time t, usually assuming linear interpolation between consecutively recorded locations. Such a trajectory dataset is considered k-anonymized if each trajectory is indistinguishable from at least k–1 other trajectories. Given the complicated nature of spatiotemporal data and the dependence of consecutive points in a trajectory, attributes (p, t) are considered both sensitive and quasi-identifiers at the same time. Under this setting, methods similar to those used for relational data can be employed to achieve anonymization.

Hoh and Gruteser's method [6] is an example of data perturbation with a goal of decreasing an adversary's certainty of correctly identifying a user. To do that, the so-called Path Perturbation algorithm creates fake intersection points between couples of non-intersecting trajectories if they are close enough. The crossing points must be generated within a specific time-window and within a user-specified radius, which indicates the maximum allowable perturbation and desired degree of privacy. Terrovitis and Mamoulis [13] proposed an approach that uses suppression. Trajectories are modeled as sequences of locations where users made transactions and an adversary is assumed to have partial knowledge of users' visited locations and their relative order, therefore an incomplete projection of the dataset. Based on this assumption the algorithm seeks to eliminate the minimum amount of locations from trajectories so that the remaining trajectories are *k*-anonymous w.r.t. an adversary's partial knowledge. Always Walk with Others (AWO) [11] is a generalization-based approach, which transforms trajectories into series of anonymized regions, while assuming adversary's partial or full knowledge of a trajectory. To achieve anonymity, the algorithm creates groups with representative trajectories and then iteratively adds to them their closest trajectories until they consist of *k* members. After that, *k* points from each anonymized region are randomly selected and connected to points similarly generated in adjacent regions in order to form *k* new trajectories. Monreale et al. [10] propose *k*-anonymization using spatial generalization of trajectories. In particular, their method finds characteristic points of trajectories and applies spatial clustering to them. The centroids of those clusters are then used for Voronoi tessellation of the area covered in the dataset dividing it into cells with at least *k* trajectories. Trajectories are formed by segments linking those cells.

Never Walk Alone (NWA) [1] and its extension Wait For Me (W4M) [2], proposed by Abul et al., follow a clustering-based approach which takes advantage of the inherent uncertainty of a moving object's location introducing the concept of (k,δ)-anonymity. An object's location at a given time is not a point, but a disk of radius δ, and the object could be anywhere inside that, so a trajectory is not a polyline, but a cylinder consisting of consecutive such disks. To achieve *k*-anonymity, each trajectory is assigned to a group of at least k–1 other trajectories using a greedy clustering algorithm. Then, the trajectories of each cluster are spatially translated so that they will all lie entirely within the same cylinder (uncertainty area) of radius δ/2. W4M is a variant of NWA that uses the time-tolerant EDR distance function [4] during the clustering phase in order to overcome the limitations of Euclidean distance. Moreover, W4M performs spatio-temporal instead of spatial translation to the trajectories. All the aforementioned approaches offer no degree of personalization since they assume that all users share the same privacy level *k* which is application-determined.

The most related to our work is the one proposed by Mahdavifar et al. [9]. It introduces the idea of non-uniform privacy requirements, where each trajectory is associated with its own privacy level indicating the number of trajectories it should be indistinguishable from. Trajectories are first divided into groups depending on their privacy level. Clusters are then created by randomly selecting a centroid and adding to the cluster the trajectories nearest to it if their EDR distance is lower than a threshold, until the maximum privacy requirement within the cluster is satisfied. If the requirements are not satisfied, groups with lower privacy levels are progressively searched for trajectories to be added to the cluster, until all the privacy requirements are met. Finally, the trajectories of each cluster are anonymized using a matching point algorithm that generates an anonymized trajectory as the cluster's representative. While this approach offers a greater degree of personalization than others, it still leads to a compulsory trade-off between privacy and quality for each user. If a trajectory has a high privacy requirement *k*, it will very likely be part of a large cluster, thus suffering from increased information loss and low data utility, since the user cannot set a 'quality' requirement.

3. PROBLEM FORMULATION

In this section, we present the formal background and definition of the *Personalized (K,Δ)-anonymity* problem in two variations. We assume that the trajectory τ of a moving object is a polyline in 3-dimensional space represented as a sequence of time-stamped locations: (p_1, t_1), (p_2, t_2), …, (p_n, t_n), $t_1 < t_2 < \ldots < t_n$. During the non-recorded time periods (t_i, t_{i+1}), we assume linear interpolation, i.e., the object moves along a straight line from p_i to p_{i+1} with a constant speed.

Following the definition adopted by [1], an uncertain trajectory buffer is defined as a cylindrical volume of diameter δ centered at an object's expected trajectory. Formally:

*Definition 1 (**uncertain trajectory**): Given a trajectory τ defined in [t_1, t_n] and an uncertainty threshold δ, τ^δ is the uncertain counterpart of trajectory τ, defined as follows: for each 3-dimensional point (p, t) in τ, its uncertainty area is the horizontal disk centered at (p, t) with diameter δ. The trajectory volume of τ^δ, denoted by Vol(τ^δ) is the union of all such disks for every $t \in [t_1, t_n]$. A possible motion curve of τ is any continuous function $f_{PMC}{}^\tau$: Time→R^2 defined on the interval [t_1, t_n], such that for any $t \in [t_1, t_n]$, the 3-dimensional point ($f_{PMC}{}^\tau(t)$, t) lies inside the uncertainty area at time t.* ■

*Definition 2 (**co-localized trajectories**): Two trajectories τ_1, τ_2, both defined in [t_1, t_n], are considered co-localized w.r.t. δ, if for each point (p_1, t) in τ_1 and (p_2, t) in τ_2, $t \in [t_1, t_n]$, it holds that the Euclidean distance $d(p_1, p_2) \leq \delta$; we write $Coloc_\delta(\tau_1, \tau_2)$ omitting the time interval [t_1, t_n].* ■

*Definition 3 (**(k,δ)-anonymous set of trajectories**): Given a set of trajectories S, an uncertainty threshold δ, and an anonymity threshold k, S is (k,δ)-anonymous iff $|S| \geq k$ and $Coloc_\delta(\tau_i, \tau_j)$ for each τ_i, $\tau_j \in S$.* ■

A dataset *D* of moving object trajectories is considered (k,δ)-anonymous if each of its members belongs to a (k,δ)-anonymity set. If *D* does not meet this requirement, then it should be transformed into a sanitized version, called D_s, which satisfies the aforementioned condition. Hence:

*Definition 4 (**(k,δ)-anonymity**): Given a dataset D of moving object trajectories, an uncertainty threshold δ, and an anonymity threshold k, (k,δ)-anonymity is satisfied by transforming D to D_s, such that for each trajectory $\tau_s \in D_s$ there exists a (k,δ)-anonymity set $S \subseteq D_s$, $\tau_s \in S$, and the distortion between D and D_s is minimal.* ■

One of the possible approaches to transform a dataset to its sanitized version is the spatiotemporal translation of the trajectory points. Distortion usually measures the difference between the original and the sanitized data. A trajectory's distortion is defined as the sum of its point-wise distances to its sanitized version. In case the trajectory is an outlier, thus removed from the anonymized dataset, the distortion is proportional to the number of the distorted moving points of the original trajectory. The total distortion caused by sanitizing the entire database is defined as the aggregation of its individual trajectories' distortion.

*Definition 5 (**trajectory distortion due to translation**): Given a trajectory $\tau \in D$ defined in [t_1, t_n] and its sanitized version $\tau_s \in D_s$, the translation distortion (TD) over τ due to its translation into τ_s is defined as:*

$$TD(\tau, \tau_s) = \begin{cases} \sum_{t \in [t_1, t_n]} d(\tau[t], \tau_s[t]) & |\tau^s| > 0 \\ |\tau| \cdot \Omega & |\tau^s| = 0 \end{cases} \qquad (1)$$

where $|\tau|, |\tau^S|$ *indicate the size (i.e., number of points) of the original and the sanitized trajectory, respectively, and Ω is a constant that penalizes distorted moving points. Summing for all trajectories, the total translation distortion over a trajectory dataset D due to its translation into* D_s *is defined as:*

$$TTD(D, D_S) = \sum_{\tau \in D} TD(\tau, \tau_S) \qquad (2)$$

■

Regarding Ω, in our experiments it corresponds to the maximum translation occurred during the anonymization process.

The problem introduced in this paper is that of achieving anonymity of a trajectory dataset, where each trajectory prescribes its own (k_i, δ_i) values, while keeping the total distortion as low as possible. The problem is formulated as follows:

*Problem 1 (**Personalized (K,Δ)-anonymity problem**): Given a dataset D of moving object trajectories, D = {*τ_1*, …,* τ_n*}, along with their respective anonymity preferences (*k_i*,* δ_i*), and a trash size threshold* $trash_{max}$*, the Personalized (K,Δ)-anonymity problem, where K = {*k_1*, …,* k_n*} and Δ = {*δ_1*, …,* δ_n*}, is to find an anonymized version of D,* D_s *= {*τ_{s_1}*, …,* τ_{s_m}*},* $0 \leq n\!-\!m \leq trash_{max}$*, where* τ_{s_i} *is a (*k_i*,* δ_i*)-anonymity version of* τ_i *and the total distortion, TTD(D,* D_s*), is minimal.* ■

In the above definition, please note that $m \leq n$, i.e., the cardinality of the output dataset D_s may be lower than that of the input dataset *D*. This is due to the fact that during the anonymization process some of the original trajectories may be moved to the trash bin, i.e., they are completely removed.

A comment on Problem 1 is that the distortion caused in the original dataset due to anonymization is not controlled since it has to do with the nature of the trajectories, as well as the values of their anonymity preferences (k_i, δ_i). Therefore, a natural variation of the above problem is that of anonymizing a database of trajectories of moving objects where each object has its own (k_i, δ_i) values, while keeping a control over the overall distortion. This problem is formulated as follows:

*Problem 2 (**Bounded Personalized (K,Δ)-anonymity problem**): Given a dataset D of moving object trajectories, D = {*τ_1*, …,* τ_n*}, along with their respective anonymity preferences (*k_i*,* δ_i*), a trash size threshold* $trash_{max}$*, and a distortion threshold* $distort_{max}$*, the Bounded Personalized (K,Δ)-anonymity problem, where K = {*k_1*, …,* k_n*} and Δ = {*δ_1*, …,* δ_n*}, is to find an anonymized version of D,* D_s *= {*τ_{s_1}*, …,* τ_{s_m}*},* $0 \leq n\!-\!m \leq trash_{max}$*, where* τ_{s_i} *is a (*k_i*,* δ_i*)-sanitized version of* τ_i *and the total distortion, Distortion(D,* D_s*)≤* $distort_{max}$*.* ■

The total distortion of the dataset when compared to its sanitized version, *Distortion(D,* D_s*)*, is a formula consisting of two factors, i.e. the distortion from the translation during the anonymization step, *TTD(D,* D_s*)*, which is calculated according to Eq.(2) along with the distortion caused from the editing phase, *TE(D)* (to be introduced in Section 5).

A special case is when the solution D_s of Problem 1 also makes a solution to Problem 2, formally: *Distortion*(*D*, D_s) ≤ $distort_{max}$; in such case, nothing extra has to be done in order to provide a solution to Problem 2. In the general case, however, where *Distortion*(*D*, D_s) > $distort_{max}$, Problem 2 can be solved by relaxing the (k_i, δ_i) constraints of those trajectories that are most responsible for the distortion caused; we call them, the most *demanding* ones, and a formulation of the demandingness of a trajectory will be defined in Section 5.

4. PERSONALIZED (*K*,*Δ*)-ANONYMITY

In this section, we present a suite of methods for publishing spatiotemporal trajectory data using the personalized (*K*,*Δ*)-anonymity. In particular, Section 4.1 describes baseline solutions for providing personalized anonymization w.r.t. different users' preferences. In section 4.2, we present an approach that provides personalized (*K*,*Δ*)-anonymity based on trajectory segmentation.

4.1 Baseline Solutions

Given a trajectory dataset *D* along with personalized privacy requirements (k_i, δ_i) for each trajectory τ_i, a baseline solution to Problem 1 consists of exploiting a state-of-the-art (*k*,*δ*)-anonymity algorithm, such NWA [1] or W4M [2], using a single, universal value for each *k* and *δ*. In order to satisfy every user's privacy requirement, it is the maximum among all k_i and the minimum among all δ_i that are assigned to the universal *k* and *δ* variables, respectively. The following algorithm, being the naïve version of our *Who-Cares-about-Others'-Privacy* (WCOP) suite of methods, illustrates this solution. As already discussed, Function *k-δ-anonymity*( ) in Line 3 of the algorithm corresponds to a state-of-the-art (*k*,*δ*)-anonymity algorithm, such as NWA or W4M.

Algorithm 1. WCOP-NV

Input: (1) a trajectory dataset, *D* = {(τ_1, k_1, δ_1), …, (τ_n, k_n, δ_n)}; (2) a trash size threshold, $trash_{max}$, (3) a maximum cluster radius threshold, $radius_{max}$
Output: A sanitized trajectory dataset, D_s = {τ_{s_1}, …, τ_{s_m}}

```
1.  $max_k \leftarrow max_i\{k_i\}$
2.  $min_\delta \leftarrow min_i\{\delta_i\}$
3.  $D_s \leftarrow$ k-δ-anonymity($D$, $max_k$, $min_\delta$, $trash_{max}$, $radius_{max}$)
4.  return $D_s$
```

In order to improve this very crude attempt of satisfying personalized (*K*,*Δ*) values, we propose an alternative approach directly using the user-specific privacy requirements and being based on clustering and translation, called WCOP-CT (where 'CT' stands for Clustering and Translation). WCOP-CT follows the general structure of W4M, consisting of two phases: a *greedy clustering phase*, which has been shown in [1] to have the best effectiveness/efficiency ratio, followed by a *spatiotemporal translation phase*, which uses Edit Distance on Real sequence (EDR) distance function [4] in order to modify each cluster to make it an anonymity set. During the first phase, a pivot trajectory is randomly selected and a cluster is formed around it by its *k*–1 unvisited closest neighbors. Then the unvisited trajectory that is farthest away from previous pivots is selected as a new pivot, and the process is repeated until clusters satisfying certain criteria. During the second phase, each cluster formed in the previous phase is transformed into a (*k*,*δ*)-anonymity set.

The input of WCOP-CT algorithm is a dataset *D* consisting of *n* trajectories τ_i along with their personalized privacy requirements (k_i, δ_i), a $trash_{max}$ value that bounds the size of trash, which contains the outliers suppressed during the clustering process (phase 1 of the algorithm) in order to improve the quality of the final output and the maximum allowable cluster radius, $radius_{max}$. The output of the algorithm is the personalized (*K*,*Δ*)-anonymized dataset D_s.

Algorithm 2. WCOP-CT

Input: (1) a trajectory dataset, $D = \{(\tau_1, k_1, \delta_1), \ldots, (\tau_n, k_n, \delta_n)\}$; (2) a trash size threshold, $trash_{max}$, (3) a maximum cluster radius threshold, $radius_{max}$
Output: A sanitized trajectory dataset, $D_s = \{\tau_{s_1}, \ldots, \tau_{s_m}\}$

```
1.  $D_s \leftarrow \varnothing$
2.  $\gamma \leftarrow$ WCOP-Clustering($D$, $trash_{max}$, $radius_{max}$)
    /* Clustering phase*/
3.  for each cluster $C \in \gamma$ do
4.      $C_s \leftarrow \varnothing$
5.      $\tau_c \leftarrow$ pivot of $C$; $\delta_c \leftarrow min_i\{\delta_i\}$ in $C$
        /* Translation phase */
6.      for each $\tau \in C$ do
7.          $\tau_s \leftarrow$ WCOP-Translation($\tau, \tau_c, \delta_c$)
8.          $C_s \leftarrow C_s \cup \{\tau_s\}$
9.      end for
10.     $D_s \leftarrow D_s \cup C_s$
11. end for
12. return $D_s$
```

In detail, WCOP-CT works as follows: the operation *WCOP-Clustering* (line 2) extracts a set of clusters γ from the original dataset. Then, for each cluster (represented by the corresponding pivot trajectory), the algorithm defines its own δ value, which is the $min_i\{\delta_i\}$ among its members (lines 3-5). All trajectories contained on the current cluster are translated by the *WCOP-Translation* operation (lines 6-7). The procedure is repeated until all trajectories of all clusters are anonymized.

The main difference between the proposed WCOP-CT and, e.g. W4M is that, whereas in W4M a pivot is selected and then invariably grouped along with its k–1 closest neighbors in order to form a cluster, in WCOP-CT each cluster has its own, non-fixed k value. It can easily be seen that this approach results in clusters of non-fixed size ranging between 2 and $max_i\{k_i\}$. In the same spirit, the spatiotemporal editing phase of WCOP-CT, called WCOP-Translation in Algorithm 2, differs to that of W4M in that there is no universal δ applied to all clusters, but each cluster is edited based on its own δ_c value, which is the $min_i\{\delta_i\}$ among its members.

The core of WCOP-CT, i.e., the clustering step under the name WCOP-Clustering(D, $trash_{max}$, $radius_{max}$) in Algorithm 2 above, is listed in Algorithm 3 below. WCOP-Clustering follows the general structure of the respective algorithm of W4M and Greedy Clustering, where W4M is based on. After forming clusters, each of them is separately processed and transformed into a (k,δ)-anonymity set, with (k,δ) being values specific to the cluster. Here we follow the approach proposed in [2], which achieves that by using the cluster's pivot as reference and editing the other trajectories so that they are co-located with it (see Section 3) and also have the same number of points as the pivot. The difference with our method is that each cluster uses its own δ value for co-localization instead of a universal value.

In detail, WCOP-Clustering iteratively selects pivot trajectories to function as centers of clusters, with pivots being selected at random from amongst the available active trajectories (Line 4). A pivot's (k_i, δ_i) values serve as the initial (k,δ) requirements of its candidate cluster (Line 6). The algorithm then successively adds to the candidate cluster the nearest unvisited neighbor of the pivot and updates the cluster's k and δ, until the cluster's size is enough to satisfy its k requirement, which equals to the maximum k_i value among its members (Lines 7-11).

Algorithm 3. WCOP-Clustering

Input: (1) a trajectory dataset, $D = \{(\tau_1, k_1, \delta_1), \ldots, (\tau_n, k_n, \delta_n)\}$; (2) a trash size threshold, $trash_{max}$, (3) a maximum cluster radius threshold, $radius_{max}$
Output: A set of clusters γ

```
1.  repeat
2.      Active ← D; Clustered ← ∅; Pivots ← ∅; Trash ← ∅
3.      while Active ≠ ∅ do
4.          $\tau_p \leftarrow$ random($\tau$) | $\tau \in$ Active
5.          $c_{\tau_p}$.size ← 1
6.          $c_{\tau_p}.k \leftarrow \tau_p.k$; $c_{\tau_p}.\delta \leftarrow \tau_p.\delta$
7.          while ($c_{\tau_p}.k > c_{\tau_p}$.size) do
8.              $c_{\tau_p} \leftarrow \{\tau_p\} \cup$ {NN of $\tau_p \in$ D – Clustered}
9.              $c_{\tau_p}$.size ← $c_{\tau_p}$.size + 1
10.             $c_{\tau_p}.k \leftarrow \max(c_{\tau_p}.k, \tau_{NN}.k)$
11.             $c_{\tau_p}.\delta \leftarrow \min(c_{\tau_p}.\delta, \tau_{NN}.\delta)$
12.         end while
13.         if $\max_{\tau \epsilon c_{\tau_p}} Dist(\tau_p, \tau) \leq radius_{max}$ then
14.             Active ← Active – $c_{\tau_p}$
15.             Clustered ← Clustered ∪ $c_{\tau_p}$
16.             Pivots ← Pivots ∪ $\{\tau_p\}$
17.         else
18.             Active ← Active – $\{\tau_p\}$
19.     end while
20.     for each $\tau \in$ D – Clustered do
21.         $\tau_p \leftarrow \mathrm{argmin}_{\tau' \in Pivots | c_{\tau'_p}.size \geq \tau.k-1,\ c_{\tau'_p} \leq \tau.\delta} Dist(\tau', \tau)$
22.         if $Dist(\tau_p, \tau) \leq radius_{max}$ then
23.             $c_{\tau_p} \leftarrow c_{\tau_p} \cup \{\tau\}$
24.         else
25.             Trash ← Trash ∪ $\{\tau\}$
26.     end for
27.     increase($radius_{max}$)
28. until |Trash |≤ |Trash_max|
29. return $\{c_{\tau_p} \mid \tau_p \in$ Pivots }
```

Once all possible clusters have been formed, the remaining unassigned trajectories are assigned to the cluster of their closest pivot, on condition that their k_i can be satisfied by the cluster's size (including themselves), their δ_i are not smaller than the cluster's current δ, and their addition will not increase the cluster's radius beyond $radius_{max}$ (Lines 20-23). If a trajectory cannot be added to any cluster without violating a condition, it is moved to the trash (Line 25). If the solution found results in trash with size larger than the $trash_{max}$ threshold, the $radius_{max}$ constraint is relaxed and the process starts again from the beginning until a solution is achieved that satisfies the $trash_{max}$ size requirement (Lines 27-28). As output, the algorithm returns only the clusters formed, excluding the suppressed trajectories implicitly (Line 29).

After the clusters have been defined, Algorithm 2 proceeds to the necessary spatiotemporal translation. Since trajectories may not be of the same size, the EDR time-tolerant distance function is responsible to minimize the necessary number of operations so as to make them indistinguishable. The goal of the *editing* operations (i.e. translate points towards pivot, remove deleted points, insert new points) that are performed to a trajectory is to make it more similar to the pivot. Algorithm 4 describes the translation procedure that is followed by WCOP-CT.

Algorithm 4. WCOP-Translation

```
Input: (1) a trajectory $\tau$, (2) cluster's pivot trajectory $\tau_c$, (3) cluster's uncertainty threshold $\delta_c$
Output: Anonymized trajectory $\tau_s$
1.  edit ← EDR_op_sequence($\tau$, $\tau_c$)
2.  $\tau_s$ ← <>
3.  i, j ← 1
4.  for all op ∈ edit do
5.      if op=remove($\tau_{c,j}$) then
6.          $\tau_s$.append(<random_point_in_circle($\tau_{c,j}.x$, $\tau_{c,j}.y$, $\delta_c/2$), $\tau_{c,j}.t$>)
7.          j ← j+1
8.      else
9.      if op=match($\tau_i$, $\tau_{c,j}$) then
10.         $\tau_s$.append(<transl($\tau_i.x$, $\tau_i.y$, $\tau_{c,j}.x$, $\tau_{c,j}.y$, $\delta_c/2$), $\tau_{c,j}.t$>)
11.         i ← i+1
12.         j ← j+1
13.     else
14.         i ← i+1
15. end for
16. return $\tau_s$
```

As a first step, the algorithm reconstructs the sequence of the required operations (Line 1). In case of a point deletion from the pivot trajectory τ_c, the algorithm instead of deleting, it creates a new point randomly inside the circle of radius $\frac{\delta_c}{2}$ around the corresponding point in τ_c (Lines 5-7). Recall that in our case each cluster has its own δ_c equal to $min_i\{\delta_i\}$ among its members. Differently, when the deletion concerns trajectory τ, the point is permanently removed (Lines 13-14). If a matching between τ and τ_c occurs, then operation *transl* is responsible to ensure that the distance between them will be equal or less than $\frac{\delta_c}{2}$. If the two points match w.r.t. the temporal dimension, then trajectory's point is transferred inside the circle to a point having the minimum distance translation from the original one. Else, the temporal coordinate value of the pivot's point is used and the point is spatially translated again inside the circle with radius equal to or less than $\frac{\delta_c}{2}$ (Lines 9-12).

4.2 Personalized (*K*,*Δ*)-Anonymity using Trajectory Segmentation

In this section, we introduce a novel approach to the problem of *Personalized (K,Δ)-anonymity*, which aims to improve upon the baseline solutions presented in the previous section by implementing trajectory segmentation, in order to increase clustering effectiveness and decrease distortion levels. A shortcoming of the two baseline solutions (WCOP-NV and WCOP-CT), which is common in all clustering methods, is that they use the trajectory as the smallest working unit. As a result, when two trajectories have some similar parts, but are significantly different on the whole, the algorithm is unable to discover and make use of those similar elements, leading to an overall increased distortion during clustering. In order to deal with this issue, our approach includes a trajectory segmentation phase, where trajectories are partitioned into sub-trajectories according to a set of privacy-aware criteria. It is these sub-trajectories that are then used as input for the anonymization stage of the algorithm that follows. While this segmentation incurs extra computational cost, it offers a distinct advantage in that it facilitates the discovery of patterns shared between parts of trajectories, which otherwise are significantly different on the whole.

The so-called WCOP-SA (where 'SA' stands for Segmenting and Anonymizing), which is presented in Algorithm 5 below, is the generic two-step method that we propose. Given a dataset *D* consisting of *n* trajectories τ_i along with their personalized privacy requirements (k_i, δ_i), in the first step the algorithm applies a trajectory segmentation process to produce the respective dataset of partitioned sub-trajectories D_p, followed by the second step that anonymizes those sub-trajectories.

Algorithm 5. WCOP-SA

```
Input: (1) a trajectory dataset, $D = \{(\tau_1, k_1, \delta_1), ..., (\tau_n, k_n, \delta_n)\}$; (2) a trash size threshold, $trash_{max}$, (3) a maximum cluster radius threshold, $radius_{max}$
Output: A sanitized trajectory dataset, $D_s = \{\tau_{s_1}, ..., \tau_{s_m}\}$.
$D_p$ ← WCOP-Segmenting($D$)
$D_s$ ← WCOP-Anonymizing($D_p$, $trash_{max}$, $radius_{max}$)
return $D_s$
```

An example of using WCOP-SA on a dataset partitioned using this method can be seen in Figure 1(d), where segments that are similar in terms of the number of their neighboring trajectories have been identified and grouped into sub-trajectories, which in turn have been assigned to appropriate clusters causing less spatiotemporal translation. WCOP-SA is by purpose generic, in that it does not strictly specify the algorithms used for segmentation and anonymization. Any algorithm of this kind can be used. However, in our experimental study (Section 6) we evaluate WCOP-SA using a trajectory-aware (Traclus [8]) vs. a neighborhood-aware segmentation algorithm (Convoys [7]) for the segmentation step, and WCOP-CT vs. WCOP-B (to be introduced in Section 5) for the anonymization step.

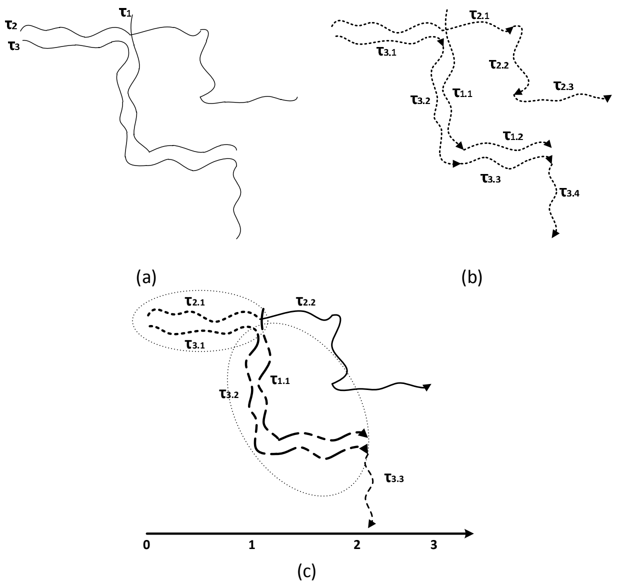


Figure 2: (a) a set of trajectories segmented by (b) Traclus and (c) Convoys partitioning-and-clustering algorithms.

Why Traclus vs. Convoys? Traclus [8] is a well-known and widely-used partitioning and clustering framework that performs density-based clustering on line segments aiming at discovering common sub-trajectories instead of grouping trajectories as a whole. During this process, trajectories are first partitioned on segmentation points representing significant changes of the trajectory's behavior (i.e. direction) by using the minimum description length principle.

Then, the directed line segments previously discovered are clustered with a variant of DBSCAN density-based clustering algorithm. However, the aforementioned approach does not properly incorporate the temporal dimension of trajectories as it spatially segments trajectories w.r.t their direction. A well-known temporal-awareness approach for clustering spatiotemporal trajectories is the one of Convoys [7]. It is a concept that uses different criteria for grouping the trajectories. A *convoy* is defined as a group of objects that has at least *m* objects, which are density-connected with respect to a distance threshold *e*, during *k* consecutive time-instants.

The difference between the two aforementioned approaches is illustrated in Figure 2. Let us assume three trajectories τ_1, τ_2, τ_3 (Figure 2(a)); when Traclus is applied for the segmentation of trajectories, the derived sub-trajectories (illustrated in Figure 2(b)) are constructed based on geometric parameters, actually, significant changes on their own direction. In contrast, Convoys performs segmentation by discovering neighboring trajectories that are moving together during a time period; as illustrated in Figure 2(c), τ_2 and τ_3 are moving together between t = 0 and t = 1 while τ_1 and τ_3 between t = 1 and t = 2, thus two convoys are discovered, which are then used for the segmentation of the trajectories to 6 sub-trajectories.

5. BOUNDED PERSONALIZED (*K,Δ*)-ANONYMITY

In order to deal with the problem of *Bounded Personalized* (*K,Δ*)-*anonymity* defined in Section 3, where the requirement is to keep anonymization distortion below a given threshold, we extent the methods presented in the previous sections by introducing dataset assessment and requirement relaxation. Since distortion is due to spatiotemporal translation, a naïve approach would be to anonymize a dataset once, identify the trajectories, which have undergone the most translation and edit them. However, a trajectory τ might be translated not due to its own (k, δ) values, but in order to be assigned to a cluster that includes very demanding members, i.e., τ's neighbors. Therefore, in order to decrease the overall distortion of a dataset, we argue that the most demanding trajectories should be detected and edited.

By intuition, high values of k and low values of δ make a trajectory demanding. As such, a metric for the demandingness of a trajectory is defined as follows:

*Definition 6 (**dataset-aware trajectory demandingness**): Given a trajectory $\tau \in D$ with privacy requirements (k,δ), its dataset-aware demandingness, $d_{dem}(\tau, D) \rightarrow [0,1]$, is defined as:*

$$d_{dem}(\tau, D) = w_1 * \frac{\tau.k}{k_{max}} + w_2 * \frac{\delta_{min}}{\tau.\delta} \tag{3}$$

where $k_{max}>1$ and $\delta_{min}>0$ correspond to the maximum k and minimum δ values of trajectories in D, respectively, and (k,δ) contribute to the overall value according to some weights, $\sum w_i = 1$.

■

Eq. (3) formulates the intuition that trajectory demandingness is proportional to k and reversely proportional to δ. k_{max} and δ_{min} are used for normalization purposes and the weights w_i are introduced in order for the importance of the two components to be controlled, according to the application scenario. For simplicity, in the rest of the paper, the two components are equally weighted, i.e., $w_1 = w_2 = ½$.

As an example, consider a dataset D consisting of 50 trajectories where k_{max} = 50 and δ_{min} = 20. Table 1 below lists the top-5 demanding trajectories of the dataset, where their demandingness has been calculated according to Eq. (3). Now assume that the overall distortion caused by the sanitization of the dataset D to D_s exceeds $distort_{max}$ threshold. The two most demanding trajectories, i.e. τ_{21}, τ_5 could be edited in order to decrease the complexity of assigning them to a cluster. Trajectory τ_{21} requires to be hidden among other 49 neighbors within an area of diameter 30 m. Similarly, trajectory τ_5, less demanding, requires other 29 neighbors within an area of diameter 20 m.

Table 1: An example of editing the most demanding trajectories

τ_i	(k_i, δ_i)	$d_{dem}(\tau_i, D)$
τ_{21}	(50,30)	0.83
τ_5	(30,20)	0.8
τ_{47}	(23,100)	0.33
τ_{15}	(23,220)	0.27
τ_7	(20,200)	0.25

How can we relax these requirements? Actually, we need a measure that indicates the degree of trajectory editing. For this purpose, we define *trajectory edit cost* as the ratio of the *(k,δ)*-editing required for the particular trajectory over the *(k,δ)*-editing required for the most demanding one. Note that for a trajectory τ, its dataset-aware demandingness can be reduced by editing its k and/or δ values.

*Definition 7 (**trajectory edit cost**): Given a trajectory $\tau \in D$ with dataset-aware demandingness, d_{dem}, defined over D, and a threshold trajectory τ_{thres}, the edit cost of trajectory τ, $0 \leq cost_{edit} \leq 1$, is defined as:*

$$cost_{edit}(\tau, D) = \tag{4}$$
$$\begin{cases} \dfrac{d_{dem}(\tau, D) - d_{dem}(\tau_{thres}, D)}{\max\limits_{\tau \in D} d_{dem}(\tau, D) - d_{dem}(\tau_{thres}, D)}, & if \max\limits_{\tau \in D} d_{dem}(\tau, D) \neq d_{dem}(\tau_{thres}, D) \\ 0 \qquad , & otherwise \end{cases}$$

where $max_{\tau \in D}\, d_{dem}(D)$ is the maximum dataset-aware demandingness among the trajectories in D. ■

As a threshold trajectory, τ_{thres}, we refer to the trajectory with the maximum acceptable demandingness in the ranking. All trajectories having a higher ranking (i.e. more demanding) are being edited. Back to Table 1, let us assume that the two most demanding trajectories need to be edited. Trajectory τ_{47} will be the threshold trajectory τ_{thres}. Thus, according to Eq. (4), the edit cost of τ_{21} equals to 1 while for τ_5 it is equal to 0,94.

The distortion caused by an edited trajectory can be measured as the number of its points multiplied by the maximum dataset translation, multiplied by the trajectory's edit cost, which indicates the required editing degree compared to the maximally edited one.

*Definition 8 (**trajectory distortion due to (k,δ) editing**): Given an edited trajectory $\tau \in D$, the contribution of trajectory τ to the overall distortion cost, distort, is defined as:*

$$distort(\tau, D) = |\tau| \cdot \Omega \cdot cost_{edit}(\tau, D) \tag{5}$$

where $|\tau|$ indicates the size of the trajectory (i.e. the number of its points) and Ω is a constant that penalizes distortion. The overall *editing distortion, DE, over a trajectory dataset D due to trajectory editing, is defined as:*

$$DE(D) = \sum_{\tau \in D} distort(\tau, D) \qquad (6)$$

■

Regarding Ω, as in Eq. (2), we define it to be the maximum translation occurred during the anonymization process.

*Definition 9 (**dataset distortion**): The total distortion of a dataset D due to its anonymization to D_s, is defined as the sum of the total distortion due to translation, TTD, and the total distortion due to editing, DE:*

$$Distortion(D, D_s) = TTD(D, D_s) + DE(D) \qquad (7)$$

■

WCOP-B ('B' stands for Bounded), which is presented in Algorithm 6 below, shows the generic concept we propose to tackle the *Bounded Personalized (K,Δ)-anonymity problem*, as it was defined in Section 3. Note that, as a first step, a user is able to apply WCOP-CT on the original dataset. The output will be an anonymized dataset along with the distortion caused during the anonymization. The user is then capable to estimate the desired distortion thus determining *distort$_{max}$*.

Algorithm 6. WCOP-B

Input: (1) a trajectory dataset, $D = \{(\tau_1, k_1, \delta_1), \ldots, (\tau_n, k_n, \delta_n)\}$; (2) a trash size threshold, *trash$_{max}$*; (3) a distortion threshold, *distort$_{max}$*; (3) an amount of editable trajectories, *step*.
Output: A sanitized trajectory dataset, $D_s = \{\tau_{s_1}, \ldots, \tau_{s_m}\}$

```
1.  for each τ ∈ D do
2.      Calculate $d_{dem}(\tau, D)$                // according to Eq. (3)
3.  end for
4.  $edit_{size} \leftarrow step$
5.  SortByDemandingness(D)
6.  repeat
7.      ResetTrajectories(D)
8.      Edited ← ∅; Trashed ← ∅; $edit_{count} \leftarrow 0$
9.      τ ← highest scoring trajectory
10.     $\tau_{thres} \leftarrow \tau_{N-edit_{size}-1}$
        / *Editing phase */
11.     while $edit_{count} < edit_{size}$ do
12.         Calculate $cost_{edit}(\tau, D)$        // according to Eq. (4)
13.         $\tau.k \leftarrow \tau_{thres}.k$
14.         $\tau.\delta \leftarrow \tau_{thres}.\delta$
15.         Edited ← Edited ∪ {τ}
16.         $edit_{count} \leftarrow edit_{count} + 1$
17.         τ ← next trajectory
18.     end while
        / * Anonymization phase */
19.     $D_s \leftarrow$ WCOP-CT(D, $trash_{max}$, $radius_{max}$)
20.     Calculate Distortion(D, $D_s$)           // according to Eq. (7)
21.     $edit_{size} \leftarrow edit_{size} + step$
22. until (Distortion ≤ $distort_{max}$ || $edit_{size} \geq |D|$)
23. return $D_s$
```

Algorithm WCOP-B-Edit-and-Anonymization works as follows: With the trajectory database and a distortion threshold *distort$_{max}$* given as input, the trajectories are first assessed in order for the algorithm to calculate the demandingness of each trajectory according to Eq. (3) (Lines 1-3). Then, the trajectories are sorted according to their demandingness, to facilitate the steps that follow (line 5). Based on the demandingness scores previously calculated, the (*k*,*δ*) values of the most demanding trajectories are edited, with *edit$_{size}$* determining the amount of editable trajectories (Lines 11-18). In more detail, the goal of the editing process that follows is to edit the most expensive trajectories so that their editing score will become equal to the threshold trajectory's editing score. Starting with the highest-scoring trajectory and continuing until *edit$_{size}$* has been reached, the editing cost of each trajectory is calculated (lines 11-12). Trajectory's k value is then decreased to the corresponding value of the threshold trajectory (lines 13). Next, the trajectory's δ is increased up to threshold trajectory's δ value (Line 14). The trajectory is then marked as 'edited', the edit-counter is updated and the next-highest-ranking trajectory selected (Lines 15-17). After the editing phase is completed, the edited dataset *D* is given as input to the WCOP-CT algorithm, which produces an anonymized dataset D_s (Line 19). The total distortion of the dataset is then calculated according to Eq.(7) (Lines 20). If the total distortion is below the distortion threshold, *distort$_{max}$*, the algorithm ends and the anonymized dataset is given as output, otherwise the portion of the dataset that is marked for editing is increased (Line 21) and the editing - anonymization phase is repeated; this loop continues until either the distortion requirement is satisfied or the entire dataset has been edited (Line 22).

It is worth to note that the method is valid for datasets consisting of either whole trajectories or segmented sub-trajectories. Therefore, it is the same algorithm that can be used in combination with either WCOP-CT or WCOP-SA (see line 19 in Algorithm 6).

Since the distortion caused by the anonymization of a dataset is heavily dependent on the original data and the dataset's privacy / quality requirements, it is possible that there will be combinations of strict distortion requirements and very demanding datasets that prohibit the discovery of a solution.

6. EXPERIMENTAL STUDY

In this section, we evaluate the effectiveness of our WCOP suite of methods for addressing the *Personalized (K,Δ)-anonymity problem* and its *Bounded* variation, as defined in Section 3. Namely, our suite consists of four algorithms: WCOP-NV, WCOP-CT, and WCOP-SA that address the first problem and WCOP-B that addresses the second problem.

We describe the experimental settings in Section 6.1. We make a base comparison between all the proposed algorithms in Section 6.2, while in Section 6.3, we briefly discuss the effects of (*k*, *δ*) parameter values. In Section 6.4, we examine the results of having first partitioned the trajectories of the dataset into sub-trajectories using dataset-aware criteria. In Section 6.5, we validate the results of using trajectory editing to relax demanding trajectories' requirements so as to decrease anonymization distortion.

6.1 Experimental Setting

In this experimental study we use a real dataset to evaluate the performance of the examined algorithms. In particular, we use a sample of GeoLife dataset [14] reporting the traces of a group of individuals monitored in Beijing, consisting of 238 trajectories.

The dataset used in our experiments is visualized in Figure 3 whereas in Table 2, we report the characteristics of the dataset, namely the number of objects – users, the number of trajectories, |*D*|, the total number of spatiotemporal points composing those trajectories, the derived average speed, the half-diagonal of the minimum bounding box of the entire space that the dataset is covering, *radius(D)*, and the duration of the dataset.

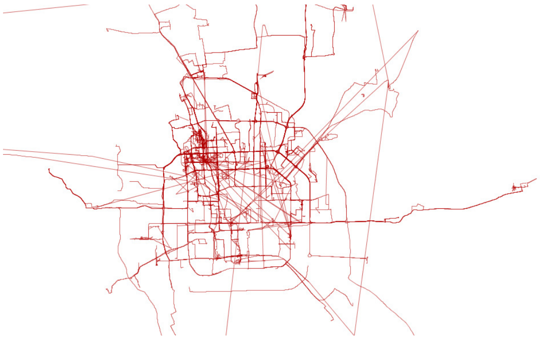


Figure 3: the trajectory dataset used in the experimental study (portion of GeoLife dataset).

Table 2: Statistics of GeoLife dataset

GeoLife	
# objects (users)	72
# trajectories, |D|	238
# spatiotemporal points	343,129
avg. speed (in m/s)	6.36
half-diagonal of entire space, radius(D) (in m)	51,982
dataset duration (in days)	1,477

δ_{max} parameter is set to 3% of *radius(D)*. $trash_{max}$, i.e., the maximum number of trajectories that can be suppressed, is set to 10% of |*D*|. $Radius_{max}$ is set equal to *radius(D)*. Finally, the tolerance thresholds of the EDR, $\Delta = \{dx, dy, dt\}$, are set as heuristic functions of δ_{max}: $\Delta = \{10 * \delta_{max}, 10 * \delta_{max}, 10 * \delta_{max} / avg\_speed\}$, where *avg_speed* is the average speed of all the moving objects in the dataset.

The experiments were performed on an Intel Xeon 2 GHz processor with 4 Gb of RAM and all the proposed algorithms were implemented in C.

6.2 Base Comparison

In this section, a base comparison between the proposed approaches is presented, in order to prove the validity of our personalized approaches. The dataset is used for this experiment with randomly generated (k,δ) requirements for each trajectory, $k \in [2, 100]$, $\delta \in [10, 1400]$. WCOP-NV finds the $\max\{k_i\}$, $\min\{\delta_i\}$ values in the dataset and uses them for all trajectories, ignoring their individual requirements, essentially replicating the way W4M works. WCOP-CT does not take universal (k,δ) values as input, instead it parses each trajectory's specific (k_i,δ_i) requirements from the dataset and uses them throughout the process. Moreover, WCOP-SA algorithms first converts the dataset into a set of sub-trajectories and then WCOP-CT is applied in order to anonymize them w.r.t. user preferences. Finally, WCOP-B improves the overall distortion when the dataset is anonymized with WCOP-CT by editing the most demanding trajectories.

Table 3 displays the results from the experiment previously described. In particular, it lists a number of useful statistics, such as the number of the input (sub-)trajectories, the number of created clusters, the number of trajectories and the number of trajectory points that ended to the trash bin, the discernibility metric [3], which measures the data quality of the anonymized trajectories, the number of created and deleted points on trajectories, the average spatial and temporal translation per trajectory, the total distortion according to Eq.(2) (for WCOP-B Eq. (7) is used), and the runtime.

In particular, *discernibility* [3] aims at measuring the quality of the sanitized data. Given a set of clusters $C_s = \{C_{s_i}, \ldots, C_{s_m}\}$ of D and the trash bin, *Trash*, it is defined as:

$$DM = \sum_{i=1}^{n} |C_{s_i}|^2 + |Trash| \cdot |D| \qquad (6)$$

Lower values of discernibility imply that more data elements are becoming indistinguishable.

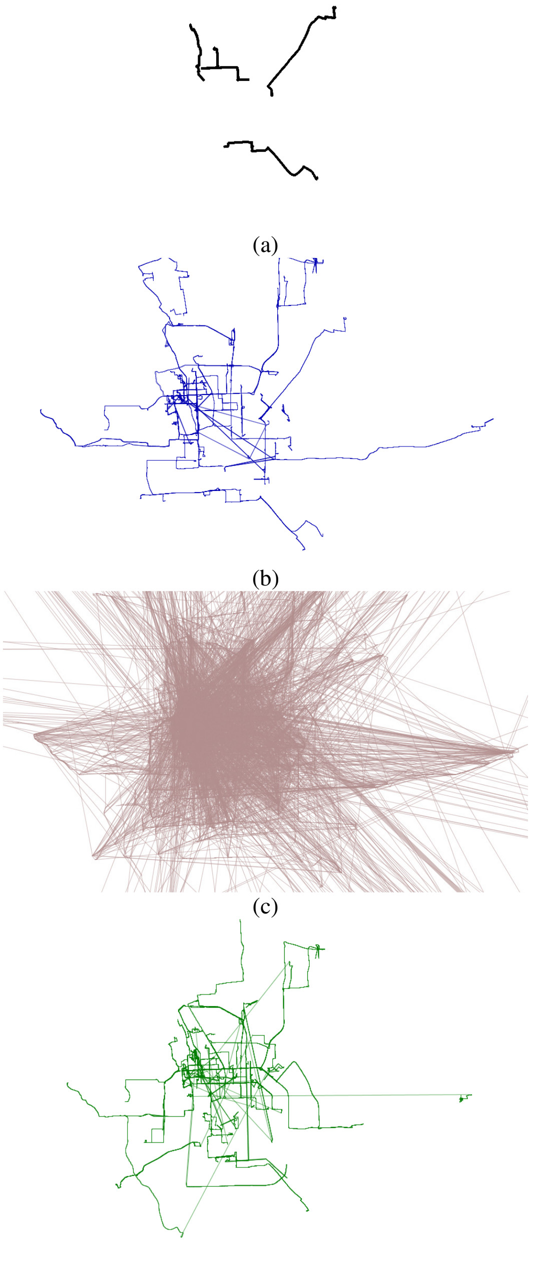
Table 3: Comparison of WCOP-NV, WCOP-CT, WCOP-SA (Traclus & Convoys), WCOP-B anonymizing the GeoLife dataset with the same parameters (k_{max}=5, δ_{max}=250)

Algor. *Stat.*	***WCOP-NV***	***WCOP-CT***	***WCOP-SA Traclus***	***WCOP-SA Convoys***	***WCOP-B***
# (sub-) trajectories	238	238	17,717	272	238
# clusters	21	55	4,412	3	51
# trajectories moved to trash	17	6	83	2	4
# points moved to trash	25,103	9,189	3,634	2,731	5,706
discernibility ($x10^3$)	19.7	2.5	1,546	40.4	2.1
# created points	14,995	41,056	176,706	75,785	47,946
# deleted points	56,086	5,118	18,704	26,321	5,509
avg. spatial translation ($x10^6$)	453	4,612	48	1,638	525
avg. temporal translation ($x10^6$)	31,633	31,244	103	33,752	30,333
total distortion ($x10^{12}$)	10.5	8.6	2.5	9.9	8.2
runtime (seconds)	30	30	120	114	414

WCOP-NV causes greater values of distortion when compared to the other approaches. The minimum distortion and the maximum discernibility metric appear when the input is a set of sub-trajectories that are segmented with the use of Traclus algorithm, thus trajectories are assigned to clusters more effectively. Moreover, 7% of the trajectories and 7% of the trajectories' points are trashed when they are anonymized by using universal *(k,δ)* privacy requirements, in contrast to WCOP-SA Traclus where the corresponding portions reaches 0.4% and 1% respectively. WCOP-B is able to decrease the overall distortion of the dataset by more than 20% when it is anonymized with WCOP-CT via editing the 6 most demanding trajectories (edit step is set to 1). Finally, runtime comparison shows that the approaches that anonymize sub-trajectories are slower than those that anonymize trajectories since a greater number of trajectories is processed. WCOP-B is even slower since every time that trajectories are edited it repeats the anonymization process until the distortion is lower than the threshold.

Based on the visualization of the aforementioned experiments as illustrated in Figure 4, we can conclude that the original trajectory dataset (see Figure 3) was better anonymized by WCOP-CT (Figure 4(b)) than by WCOP-NV (Figure 4(a)).

Clearly, WCOP-NV was not able to maintain the trend of the original trajectories. due to the reduced number of the created clusters. WCOP-CT and WCOP-SA-Convoys (Figure 4(d)) better preserved the pattern of the original trajectories. WCOP-SA-Traclus (Figure 4(c)) reports dense sanitized trajectories due to the segmentation of the original dataset that increased its size by 99%. Thus, we can argue that the result is expected.


(a)

(b)

(c)

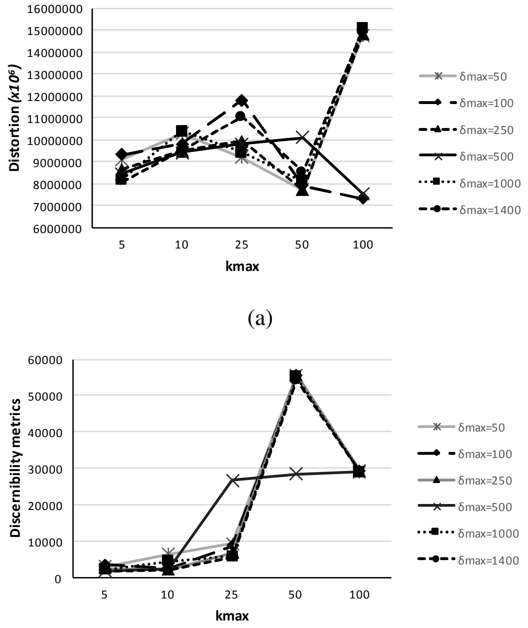
(d)

Figure 4: Anonymized trajectories by (a) WCOP-NV; (b) WCOP-CT; (c) WCOP-SA-Traclus; (d) WCOP-SA-Convoys.

6.3 The Effect of (K,Δ) Parameters

In this section, we examine the effects of using varying combinations of (K,Δ) values with respect to the total information distortion caused by the anonymization. Each trajectory's (k, δ) requirements are randomly generated, with $k \in [2, k_{max}]$, $\delta \in [10, \delta_{max}]$. As mentioned above, δ_{max} parameter has been set to 3% of the dataset bounding rectangle's radius. The k_{max} and δ_{max} variables are varying on each iteration with k_{max}={5, 10, 25, 50, 100} while δ_{max}={50, 100, 250, 500, 1000, 1400}.

Focusing on WCOP-CT, Figures 5(a) and 5(b) provide a visual representation of the total distortion and the discernibility metric, respectively, for different combinations of (k_i, δ_i). It is clear that WCOP-CT is affected from the changes both in k_{max} and δ_{max} parameters. However, there is a point in Figure 5(a) where while the distortion decreases with the increase of k_{max}, reaching the minimum values when $k=50$, a sudden increase appears when $k=100$. This is due to the fact that the number of the trash size increases up to a point that exceeds trash$_{\text{max}}$. When this occurs, $radius_{max}$ is enlarged in order to cluster more trajectories thus the number of the removed trajectories is shrinked but trajectories are spatially translated even more. This trend is obvious at the overall distortion and the discernibility metric when k_{max}=25 (Figures 5(a) and 5(b)).



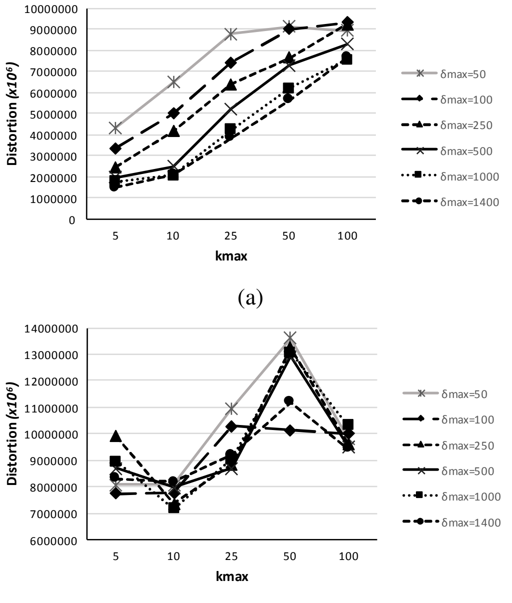
(a)

(b)

Figure 5: WCOP-CT: (a) total distortion and (b) discernibility for different combinations of (k_{max}, δ_{max}).

6.4 The Effect of Trajectory Partitioning

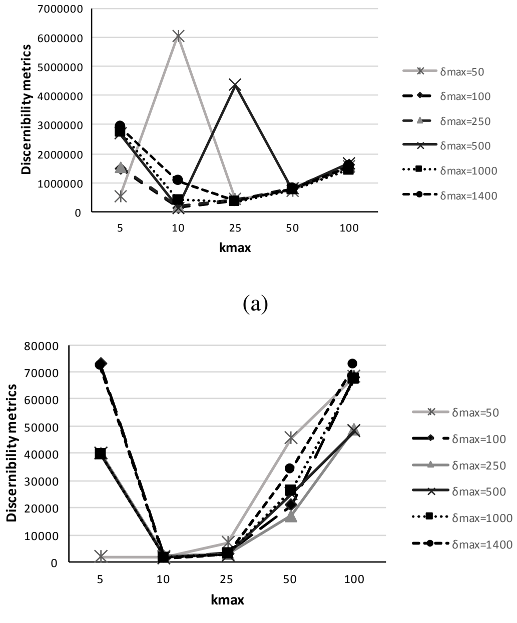
In this section, we validate WCOP-SA algorithm. In particular, we compare the effects of using WCOP-CT with two different inputs of the GeoLife dataset, i.e. trajectories after being segmented into sub-trajectories using either Traclus [7] or Convoys [6]. Regarding k and δ, they were again randomly generated with $k \in [2, k_{max}]$ and $\delta \in [10, \delta_{max}]$.



(a)

(b)

Figure 6: total distortion for different combinations of (k_{max}, δ_{max}) using (a) WCOP-SA-Traclus; (b) WCOP-SA-Convoys.



(a)

(b)

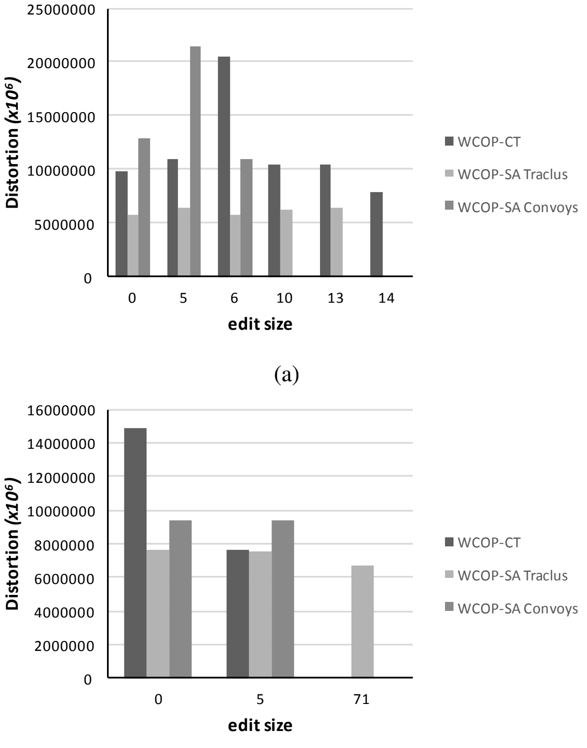
Figure 7: discernibility for different combinations of (k_{max}, δ_{max}) using (a) WCOP-SA-Traclus; (b) WCOP-SA-Convoys.

As we observe in the base comparison (see Table 3), partitioning the dataset into sub-trajectories results in very high discernibility, caused by the significantly higher number of clusters. Moreover, the segmentation of trajectories also appears to cause substantially decreased information distortion especially when they were partitioned with the Traclus algorithm. Figures 6(a) and 6(b) illustrate the total distortion in both approaches which rises as the value of k_{max} increases. Discernibility metrics in Figure 7(a) and 7(b) reports that the data quality is maintained either in lower or in higher values of k_{max}. It is worth noting that WCOP-SA with Traclus manages to increase the average quality by 99% and decrease the average total distortion by 43% compared to the corresponding average values of WCOP-CT. Similarly, WCOP-SA when using the Convoys algorithm increases the average data quality by 31% and decreases the distortion by 2%.

6.5 The Effect of Trajectory Editing

In the final part of our experimental study, we examine the effects of trajectory editing based on the algorithm outlined in Section 5. Two different versions of GeoLife dataset are applied in this set of experiments, using different ranges of randomly assigned *(k,δ)* values, i.e. [25 , 500] and [100 , 1400], in order to examine the effect of edit on the final result and how privacy requirements can influence it. Secondly, in order to examine the effects of trajectory editing on datasets of whole trajectories and on datasets consisting of segmented sub-trajectories, we apply WCOP-B in both types of data.

Figure 8(a) illustrates the effects of editing various numbers of trajectories for a dataset that corresponds to medium demanding users. In contrast, Figure 8(b) depicts the respective outcome but for much more demanding users. It is obvious that most of the approaches decreased 10% of their distortion by only editing the top-5 demanding trajectories apart from WCOP-SA-Traclus due to the increased number of sub-trajectories.



(a)

(b)

Figure 8: WCOP-B: distortion for varying edit size values where (a) k_{max} = 25 and d_{max} = 500; (b) k_{max} = 100 and d_{max} = 1400.

It is not only that distortion changes in a non-monotone as edit size increases; we also observe that it can actually increase as edit size increases. This is due to the fact that each edited trajectory incurs a distortion penalty, which grows proportionally to the edit-size. However, the distribution of demanding trajectories across the clusters and the distribution of *(k,δ)* values in the dataset significantly influence the degree to which relaxing additional trajectories' requirements affect the clustering and anonymization phases. Therefore, higher percentage of edited trajectories does not guarantee decreased distortion, indicating that there exists an 'optimal' edit-size value, where distortion is the minimum possible.

7. CONCLUSIONS

In this paper, we proposed a novel approach for anonymizing trajectories called Personalized *(K,Δ)*-Anonymity, which uses user-specific privacy requirements. Based on this framework, we have developed WCOP-CT algorithm, which takes advantage of user-specific *(k,δ)* requirements in order to assign trajectories to clusters of minimal size, so as to avoid over-anonymization, increase data quality and decrease distortion. Expanding upon that framework, we made use of dataset-aware trajectory segmentation, in order to further improve our approach's effectiveness, by partitioning trajectories to sub-trajectories that are more easily assignable to clusters. Additionally, we examined the concept of Bounded *(K,Δ)*-Anonymity, whereby there is a threshold to the acceptable distortion caused by the anonymization process, and proposed methods for trajectory assessment and editing by relaxing the requirements of the most demanding trajectories without editing the spatiotemporal data.

To show the effectiveness of our methods, we have performed experiments using the GeoLife dataset. Our personalized anonymity approach has been shown to significantly increase to the overall quality and to decrease of the total distortion of the anonymized datasets, while it has also been demonstrated that trajectory segmentation can improve data quality even further. Experimental results also show that our trajectory assessment and editing algorithms perform very well towards the goal of decreasing data distortion without altering the trajectories' spatiotemporal information itself.

Overall, we argue that we have provided a novel approach in mobility data anonymization. Using our WCOP suite of techniques, data analysts are able to preserve the quality of anonymized datasets taking advantage of user-specific privacy requirements combined with methods such as segmentation and trajectory editing. However, there is a number of points, such as sensitivity to *(k,δ)* values distribution, replacement of greedy clustering with a more sophisticated clustering method, sensitivity to segmentation method and alternative trajectory assessment and editing methods, which deserve further study in order to expand and improve upon the framework presented here.

8. ACKNOWLEDGMENTS

The publication of this paper has been partly supported by the University of Piraeus Research Center.

9. REFERENCES

[1] Abul, O., Bonchi, F., Nanni, M. (2008). Never walk alone: Uncertainty for anonymity in moving objects databases. *In Proceedings of International Conference on Data Engineering*, *ICDE*, pp. 376-385.

[2] Abul, O., Bonchi, F., Nanni, M. (2010). Anonymization of moving objects databases by clustering and perturbation. *Information Systems*, *35*(8), pp. 884-910.

[3] Bayardo, R. J., Agrawal, R. (2005). Data privacy through optimal *k*-anonymization. *In Proceedings of the International Conference on Data Engineering*, pp. 217-228.

[4] Chen, L., Özsu, M. T., Oria, V. (2005). Robust and fast similarity search for moving object trajectories. *In Proceedings of ACM SIGMOD International Conference on Management of Data,* pp. 491-502.

[5] Chow, C. Y., Mokbel, M. F. (2011). Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter*, *13*(1), pp. 19-29.

[6] Hoh, B., Gruteser, M. (2005). Protecting location privacy through path confusion. *In Proceedings of SecureComm,* pp. 194-205.

[7] Jeung, H., Yiu, M. L., Zhou, X., Jensen, C. S., Shen, H. T. (2008). Discovery of convoys in trajectory databases. *In Proceedings of the VLDB Endowment*, 1(1), pp. 1068-1080.

[8] Lee, J. G., Han, J., & Whang, K. Y. (2007). Trajectory clustering: a partition-and-group framework. *In Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 593-604.

[9] Mahdavifar, S., Abadi, M., Kahani, M., Mahdikhani, H. (2012). A clustering-based approach for personalized privacy preserving publication of moving object trajectory data. *In Proceedings of Network and System Security*, pp. 149-165.

[10] Monreale, A., Andrienko, G. L., Andrienko, N. V., Giannotti, F., Pedreschi, D., Rinzivillo, S., Wrobel, S. (2010). Movement data anonymity through generalization. *Transactions on Data Privacy*, *3*(2), pp. 91-121.

[11] Nergiz, M. E., Atzori, M., Saygin, Y. (2008). Towards trajectory anonymization: a generalization-based approach. *In Proceedings of the ACM International Workshop on Security and Privacy in GIS and LBS, SIGSPATIAL* pp. 52-61.

[12] Sweeney, L (2002) *k*-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge Based Systems*, 10(5), pp. 557-570.

[13] Terrovitis, M., Mamoulis, N. (2008). Privacy preservation in the publication of trajectories. *In Proceedings of International Conference on Mobile Data Management MDM*, pp. 65-72.

[14] Zheng, Y., Xie, X., Ma, W.-Y. (2010). GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Eng. Bull.*, 33(2), pp. 32–39.