Expert Systems With Applications 213 (2023) 118980

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A compression strategy for an efficient TSP-based microaggregation

Armando Maya-López [a], Antoni Martínez-Ballesté [a], Fran Casino [a,b,*]

[a] Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Avinguda dels Països Catalans, 26, 43007 Tarragona, Spain
[b] Information Management Systems Institute, Athena Research Centre, Artemidos 6, Marousi 15125, Greece

ARTICLE INFO

Keywords:
Statistical disclosure control
Microaggregation
Data privacy
Travelling Salesman Problem
Data protection
k-anonymity

ABSTRACT

The advent of decentralised systems and the continuous collection of personal data managed by public and private entities require the application of measures to guarantee the privacy of individuals. Due to the necessity to preserve both the privacy and the utility of such data, different techniques have been proposed in the literature. Microaggregation, a family of data perturbation methods, relies on the principle of *k*-anonymity to aggregate personal data records. While several microaggregation heuristics exist, those based on the Travelling Salesman Problem (TSP) have been shown to outperform the state of the art when considering the trade-off between privacy protection and data utility. However, TSP-based heuristics suffer from scalability issues. Intuitively, methods that may reduce the computational time of TSP-based heuristics may incur a higher information loss. Nevertheless, in this article, we propose a method that improves the performance of TSP-based heuristics and can be used in both small and large datasets effectively. Moreover, instead of focusing only on the computational time perspective, our method can preserve and sometimes reduce the information loss resulting from the microaggregation. Extensive experiments with different benchmarks show how our method is able to outperform the current state of the art, considering the trade-off between information loss and computational time.

1. Introduction

The advent of the Big Data era paves the way for generating, storing, analysing, and exploiting large amounts of data. The so-called data science has emerged as a significant discipline, merging fields such as mathematics and computer sciences. Technology promotes the successful integration of data mining and other analysis techniques into the business intelligence software of companies and governments.

However, since a significant amount of microdata (observation data collected on an individual object) is obtained from daily-life activities (e.g. bank transfers, power consumption, location tracking), individuals' right to privacy is becoming an urgent concern.

Data regulations (European Commission, 2018) focus on the rights of citizens regarding their own data. Hence, for instance, data cannot be transferred to third parties without the individuals' consent. Having privacy preservation in mind and aiming at hiding the identity of respondents, regulations emphasise the use of techniques like pseudonymisation. Essentially, identifiers in the microdata set, i.e., the attributes that unambiguously identify the respondent (such as name and surname, social security number) are replaced by a pseudonym. Although there are various methods for implementing pseudonymisation (e.g., from counters to random numbers, including the use of cryptographic functions), this process is not straightforward for a variety of reasons. On the one hand, the correspondence between identifiers and pseudonyms must be kept separately and is naturally subject to organisational measures to guarantee that such correspondence is inaccessible to unauthorised users. On the other, despite using pseudonyms, several attributes that remain in the microdata can be combined and linked with some external information to re-identify individuals with records in the data set. As a result, sensitive data (e.g. bank funds, health status) might be disclosed from the supposedly protected microdata, because such quasi-identifier attributes are not being properly tackled.

Statistical Disclosure Control (SDC) (Willenborg & de Waal, 2001) attempts to protect individuals' right to privacy by applying off-the-shelf methods to the microdata sets. Such methods typically result in

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Avinguda dels Països Catalans, 26, 43007 Tarragona, Spain.
E-mail addresses: armando.maya@estudiants.urv.cat (A. Maya-López), antoni.martinez@urv.cat (A. Martínez-Ballesté), franciscojose.casino@urv.cat (F. Casino).

https://doi.org/10.1016/j.eswa.2022.118980
Received 10 July 2022; Received in revised form 29 September 2022; Accepted 3 October 2022
Available online 10 October 2022

some degree of data modification (or perturbation). Thus, the challenge for SDC is to maximise privacy protection whilst minimising the information loss due to data modification.

Contribution and Plan of the Article: The Travelling Salesman Problem (TSP) (Shmoys, Lenstra, Kan, & Lawler, 1985) (i.e., finding the shortest Hamiltonian path over a graph) has been extensively used in different types of applications (Osaba, Yang, & Del Ser, 2020; Zaidi, 2020), yet it has been much less explored in the SDC field due to its computational cost (Heaton & Mukherjee, 2011; Maya-López, Casino, & Solanas, 2021). In this article, we present a method to improve the efficiency of microaggregation (i.e. a family of SDC techniques) that utilises the TSP to create a path over the multivariate records. In addition to exponentially reduce the time required for the microaggregation process, our method is able to protect the natural data distribution of the dataset, thus leading to better grouping strategies than the original TSP methods. In other words, our method is able to reduce not only the computational time, but also the error introduced by microaggregation. To the best of our knowledge, this is the first time that an optimisation method for TSP-based microaggregation heuristics (i.e. optimising both the computational time and the quality of the groups) is presented in the literature. The remainder of the article is organised as follows: Section 2 provides the reader with an overview on microaggregation and statistical disclosure control (SDC). Section 3 analyses related work and underlines the scope and novelty of our technique. Section 4 describes our proposal, which is later tested and compared with classical and state-of-the-art microaggregation methods in Section 5. Section 6 analyses the benefits and limitations of our approach. Section 7 concludes the article with some final remarks.

2. Background on microaggregation

Microaggregation is a group of SDC techniques that pursue protecting individuals' privacy by achieving the k-anonymity property, which stands out among the most popular SDC techniques. In a nutshell, data is modified in a way that in the protected microdata set every combination of quasi-identifiers can be indistinctly matched to at least k individuals (Samarati & Sweeney, 1998). To achieve k-anonymity:

- In a first step, each record in the original microdata set is assigned a group or cluster, so that records in the same cluster are similar. This grouping is called the k-partition.
- In a second step, each group is aggregated: each record in the group is substituted by, typically, the centroid (e.g., the mean in case of numerical data).

Recalling the concept of balancing between privacy protection and information loss, for the released dataset to remain useful, the changes due to microaggregation must be minimised. The so-called *optimal k-partition* is the one that maximises within-group homogeneity and hence minimises information loss. In this sense, microaggregation can be regarded as an optimisation problem.

In order to assess the performance of a microaggregation technique, two aspects must be considered: on the one hand, the quality of the k-partition obtained (in terms of maximising within-group homogeneity) and, on the other, the computational cost of the method.

Regarding quality, maximising within-group homogeneity is related to finding a k-partition that, for each group, minimises the distance between its records and the centroid. A general approach for benchmarking microaggregation techniques is the sum of square errors (SSE):

$$\mathrm{SSE} = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)' \tag{1}$$

, where $x_{i,j}$ is the jth record in group i, and $\bar{x}_i$ is the centroid of group i.

Besides SSE, another measure for analysing the performance of a microaggregation method is information loss (IL). We express the IL in percentage by using the following equation:

$$\mathrm{IL} = \frac{\mathrm{SSE}}{\mathrm{SST}} \times 100 \tag{2}$$

, where SST is the total sum of squares (SST), an upper limited on the partitioning that is obtained as follows:

$$\mathrm{SST} = \sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})' \tag{3}$$

, where x_i is the ith record in D and $\bar{x}$ is the average record of the entire dataset. Note that when no microaggregation is performed, there is no loss of information: the SSE for a hypothetical 1-partition is 0 and, hence, IL = 0/SST.

Regarding the computational cost, the optimal k-partition for a single attribute (i.e. univariate microdata) can be computed in polynomial time, as described by Hansen and Mukherjee (2003). However, the data to be microaggregated is, in general, multivariate since more than one attribute must be microaggregated: in this case, finding the optimal k-partition is NP-hard (Domingo-Ferrer, Sebé, & Solanas, 2008). For this reason, a number of proposals for microaggregation aim approaching the optimal k-partition using algorithms that run in polynomial time. Some of these methods pursue mapping multivariate microdata as they were univariate, typically by ordering these records (i.e. creating a path over the records) and then using the optimal univariate microaggregation described in Hansen and Mukherjee (2003).

3. Related work

In this section, we first recall different approaches to microaggregation. Moreover, since in the Big Data era microaggregation must cope with very large datasets, we also address some techniques to palliate computational efforts.

3.1. Microaggregation techniques

Aiming at providing a classification, we can distinguish between fixed and variable-size microaggregation. The former yields k-partitions where all groups have size k, except possibly one group, which has size between k and $2k-1$ when the number of records in the data set is not a multiple of k. On the contrary, variable-size heuristics yield k-partitions where all groups have sizes between k and $2k-1$. In addition, some of the methods take advantage of the Hansen and Mukherjee technique for optimal microaggregation of univariate data.

One of the best known methods is the *Maximum Distance to Average Vector* (MDAV) (Domingo-Ferrer, Martinez-Ballesté, Mateo-Sanz, & Sebé, 2006). This method iteratively creates groups with k records considering the records furthest from the data centroid. Laszlo et al. proposed approaches similar to MDAV: the *Centroid-Based Fixed Size method* (CBFS) (Laszlo & Mukherjee, 2005), and the improvements based on kd-tree neighbourhood search, such as KD-CBFS and KD-CBFSapp (Solé, Muntés-Mulero, & Nin, 2012). Chang, Li, and Huang (2007) proposed the *Two Fixed Reference Points* (TFRP) method: in this case, the two most separate points of the dataset at each loop are used as references to create groups. Next, to decrease IL in the second phase, TFRP generates variable-size groups by removing the less homogeneous groups. *Differential Privacy-based microaggregation* (Yang, Ye, Fang, Wu, & Wang, 2020), develops a variant of the MDAV algorithm that uses the correlations between attributes to select the minimum noise required to obtain the desired level of privacy. *Variable MDAV* (V-MDAV) (Solanas & Martínez-Ballesté, 2006) is a variable group-size method based on the MDAV that allows clusters to better adapt to the data and reduce the SSE. Microaggregation based on minimum spanning trees (Laszlo & Mukherjee, 2005), aimed at creating graphs that can be pruned in accordance with each node's weights to create the clusters. *Density-Based*

Algorithm (DBA) (Lin, Wen, Hsieh, & Chang, 2010), first forms clusters in density-descending order, and then postprocesses these clusters in reverse order. Finally, *Group Selection based on sequential Minimisation of SSE* (GSMS) method (Panagiotakis & Tziritas, 2011), optimises the IL by rejecting the candidate cluster that minimises the ongoing SSE of the remaining records.

Other proposals have focused on the efficiency of the microaggregation technique. For instance, the *Fast Data-oriented Microaggregation* (FDM) (Mortazavi & Jalili, 2014) efficiently protects large multivariate numerical datasets for multiple successive values of k.

3.2. Methods based on optimal univariate microaggregation

A number of methods are built upon the Hansen and Mukherjee algorithm for optimal univariate microaggregation (HM). In a nutshell, they work as follows: addressing the multivariate records in the dataset (with p columns or variables) as if they were points in a p-dimensional space, first, use a technique to create a path traversing all the records. This path is a permutation of the records in the dataset, which are now reduced to univariate data. Afterwards, use this permutation to feed the HM technique, that will create the k-partition. This technique is known as the *Multivariate Hansen–Mukherjee* (MHM).

For example, the IMHM method (Mortazavi, Jalili, & Gohargazi, 2013) is based on the latter. Moreover, in Domingo-Ferrer et al. (2006) a grouping method that combines several approaches such as *Nearest Point Next* (NPN-MHM), MDAV-MHM, and CBFS-MHM is proposed.

Finally, *TSP-based microaggregation* focus on using the Travelling Salesman Problem heuristics to generate an ordered sequence in $\mathbb{R}^n$ of the records in a dataset. Such sequence can be used to create a k-partition (by using e.g., the MHM method). Heaton and Mukherjee (2011) used the TSP tour optimisation heuristics (e.g., 2-opt, 3-opt) to refine a path created with the information of a multi- variate microaggregation method (e.g., MDAV, MD, CBFS). Note that finding an optimal solution to the TSP is known to be NP-hard. In Maya-López et al. (2021) used a variety of off-the-shelf heuristics that find sub-optimal TSP solutions to provide HM with an ordering of records. Hence, they did not consider using a multivariate microaggregation method as a pre-processing step. Maya-López et al. showed that this new approach decreased the computational time whilst preserving data utility, outperforming the state of the art. For more about microaggregation, the interested readers can refer to Fayyoumi and Oommen (2010), Zigomitros, Casino, Solanas, and Patsakis (2020).

3.3. Dataset reduction strategies for microaggregation

The computational cost of the heuristics grows in runtime with the size of the dataset on which we want to operate. Consequently, when dealing with large datasets or using computationally demanding methods, techniques like dataset splitting (Shirkhorshidi, Aghabozorgi, Wah, & Herawan, 2014) or dimensionality reduction (Liew et al., 2016) must be contemplated.

Notwithstanding, these approaches entail several shortcomings, such as the "noise" introduced by dimensionality reduction methods when dealing with high-dimensional data (Casino, Patsakis, & Solanas, 2019; Indyk & Motwani, 1998), and the fact that splitting strategies (and hence the risk of grouping similar records into different subsets) may have a great impact on the usability of data and their statistical properties (Casino, Domingo-Ferrer, Patsakis, Puig, & Solanas, 2015; Solanas, González-Nicolás, & Martínez-Ballesté, 2010).

A baseline strategy for dataset splitting is to use clustering algorithms to generate smaller subsets. Nevertheless, clustering algorithms such as K-means may create partitions that separate the records erroneously according to data distribution, thus introducing noise and dramatically hindering the quality of the groups, as seen in Fig. 1.

Monedero, Mezher, Colomé, Forné, and Soriano (2019) proposed a constrained partitioning strategy that applies efficiently only to structured data. This constraint is partially solved in Solanas et al. (2010) due to a 2-step microaggregation process to protect groups, which allows the partitioning of a dataset regardless of the data it contains. More concretely, aiming at avoiding the split of natural clusters, the authors of the work presented in Solanas et al. (2010) applied MDAV in a two-step partitioning strategy. First, they used MDAV with a low value of k to ensure that close elements will not be separated; next, they generated partitions by using a higher value of k over the dataset created in the first step. While this strategy may be efficient for clustering-based algorithms, it is not efficient for TSP-based methods since they require all the dataset records to compute the optimal path by exploring all the possible connections.

4. Our proposal

The TSP is a well-known NP-Hard problem in the literature, whose complexity depends on the amount of "cities" (in our case, the records in the dataset) considered to compute the path traversing all records (i.e. the Hamiltonian path). This is a common step of all TSP-based microaggregation heuristics, as extensively reported in Maya-López et al. (2021).

In the words of the researchers who designed an implementation for solving the TSP, "At the current time, our notion of very large refers to problems having over 10,000 cities" (Applegate, Bixby, Chvatal, & Cook, 2006). Therefore, a strategy should be considered to calculate this type of dataset with more than 10,000 records. In this article, however, we show that optimisation strategies can be used in both small and large datasets according to the application scenario, contrary to the assumption stated above. Moreover, in addition to focusing only on the computational time perspective, our optimisation strategy can preserve, and sometimes improve, the quality of the clusters in a microaggregation setting. In what follows, we describe our proposal to leverage the performance of TSP-based microaggregation, and we analyse the benefits of the group protecting strategies in the context of TSP methods. Note that our method is designed in a way that can accommodate different compression strategies and is compatible with any TSP heuristic, enhancing its usability.

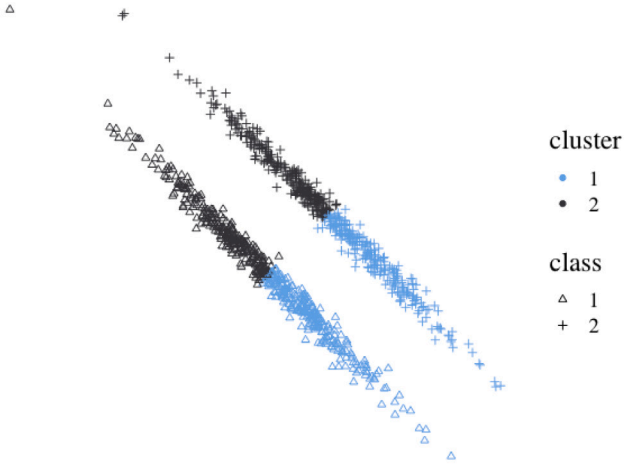
4.1. A compression strategy for efficient, TSP-based microaggregation

Algorithm 1 describes our compressed, TSP-based microaggregation process, to microaggregate a dataset D with r records and p columns, with a privacy parameter k (i.e., to create a k-anonymous version of dataset D). Essentially, it is divided into four phases:

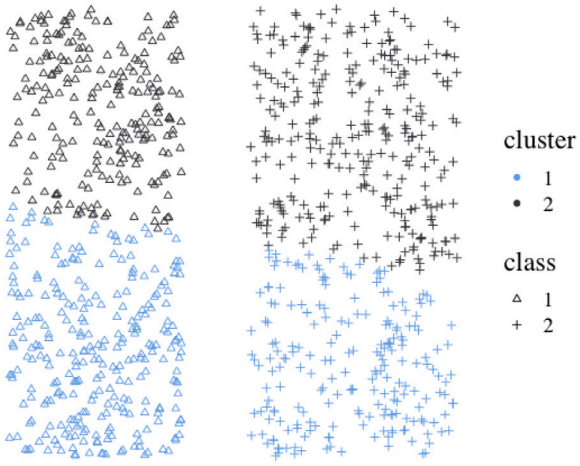
- *Phase 1. Compression.* A compressed version D_{comp} of the dataset is generated.
- *Phase 2. TSP Path Finding.* The TSP is performed to find a Hamiltonian path HP_{comp} over the compressed dataset, regarded as a graph.
- *Phase 3. Decompression.* It consists of using HP_{comp} to create a new Hamiltonian path HP_{dec} over the original dataset.
- *Phase 4. Microaggregation.* The Multivariate Hansen and Mukherjee technique is applied using the H_{dec} path to create the k-partition.

Next, we focus on some details of the algorithm. As an initial step and, to avoid bias towards higher magnitude variables, D is standardised (Algorithm 1: line 2). To this end, each value of the dataset is subtracted the average of its column and divided by the standard deviation of this column.

The first step of Phase 1 (Algorithm 1: line 3) consists in obtaining C_c, a c-partition of D_{std} using a microaggregation method m with a cardinality constraint parameter c (i.e., the compression ratio). Thus, $C_c = \{c_1, c_2, \ldots, c_{r/c}\}$ describes the clusters in the c-partition (e.g., $c_3 = \{7, 32, 94\}$ indicates that cluster number three is composed by rows 7, 32 and 94 of D).



(a) Non zero covariance dataset



(b) Uniform distribution dataset

Fig. 1. Example of K-means inefficient partition strategies over two different datasets.

In the second step of Phase 1 (Algorithm 1: line 4), the compressed dataset D_c is built, using the set of r/c centroids C_c generated in the previous step. Following the previous example, the third row in D_c is the centroid of cluster c_3, i.e., the average of records 7, 32, and 94.

Phase 2 consists of finding a Hamiltonian path over D_c. First (Algorithm 1: line 5), we model D_c as a complete graph $G(N, E)$, where we assume that each row of D_c is represented by a node $n_i \in N$ and each edge $e_{ij} \in E$ represents the Euclidean distance between n_i and n_j. Thus, we have a set of nodes $N = \{n_1, n_2, \ldots, n_{r/c}\}$ each representing rows of the compressed microdata set in a multivariate space $\mathbb{R}^p$. Next (Algorithm 1: line 6), we apply a TSP path construction heuristic over G to create a Hamiltonian path HP_C, i.e. a permutation ($\Pi^N = \{\pi_1^N, \pi_2^N, \ldots \pi_{r/c}^N\}$) of the nodes in N, which, *de facto* determines a specific order.

Phase 3, aims at creating a new Hamiltonian path HP_D, but in this case considering all the records/nodes of D_{std}, whilst preserving the specific order determined in HP_C. This process consists of iterating all the nodes of HP_C and, for each node $HP_c(i)$, insert as nodes in HP_D all the records its corresponding cluster in the c-partition C_c (Algorithm 1: line 9). The order of insertion of each cluster's records is determined by the distance to dataset's centroid according to D_{std} (Algorithm 1: line 8).

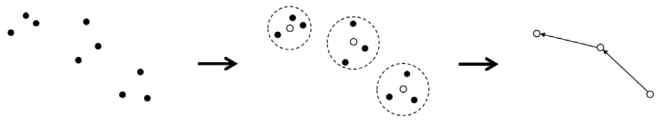
After this decompression process, the result is a new HP_D that includes the permutation of all elements existing in the original dataset.

Finally, in Phase 4 the original dataset is microaggregated. HP_D is used as input to the MHM method, together with the privacy parameter k, to create the k-partition (Algorithm 1: line 11). It returns the optimal univariate k-partition of D_{std}, which is used to build the microaggregated dataset, i.e. each record in the group is substituted by the group's centroid (Algorithm 1: line 12).

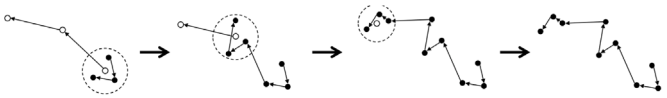
For the sake of clarity, we provide an overview of the compression and the decompression steps in Fig. 2

4.2. Path length and microaggregation

One of the primary keys supporting the feasibility of our method is the fact that a shorter path length does not guarantee the creation of better grouping strategies than using longer paths, as observed in the literature (Climer, Zhang, & Joachims, 2006). The latter becomes more evident when applying an optimisation method such as MHM, as noted later in Section 5.2. In fact, given a strategy that protects the groups according to some parameters, the resulting path may generate a more appropriate grouping strategy for microaggregation. An example of the latter is illustrated in Fig. 3. As it can be observed, paths b and c are longer than path a, mainly due to the long diagonal connection emerging from their rightmost node. Despite that, we can observe that path b can protect groups, especially for a cardinality value $k > 4$, in a



(a) Original records (left), clusters and their corresponding centroids (center), and resulting Hamiltonian Path (right).



(b) From left to right, decompression of each centroid by replacing it with the records of the original cluster. Inner nodes are ordered according to their distance to the dataset's corresponding centroid.

Fig. 2. Illustration of the compression (a) and the decompression (b) processes.

Algorithm 1 Function Compress-TSP-MHM that microaggregates the dataset D.

function COMPRESS-TSP-MHM(microdata set **D**, microaggregation method **m**, TSP method **t**, privacy parameter **k**, compression ratio **c**)
  $D_{std} \leftarrow standardizeDataset(D)$

  // Phase 1: Compression
  $C_c \leftarrow$ microaggregation(D_{std}, m, c)
  $D_{comp} \leftarrow$ getCentroids(C_c, D_{std})

  // Phase 2: TSP Path Finding
  $G \leftarrow$ createGraph(D_{comp})
  $HP_{comp} \leftarrow$ computeTSP(G)

  // Phase 3: Decompression
  for $i = 1$ to $length(HP_{comp})$ **do**
    $c \leftarrow sortRecords(C_c(HP_{comp}(i)), D_{std})$
    $HP_{dec} \leftarrow addRecords(c)$
  end for

  // Phase 4: Microaggregation
  $C_k \leftarrow$MHM(HP_{dec}, D_{std}, k)
  $M \leftarrow$ buildMicroaggregatedDataSet(C_k, D)
  return M
end function

more efficient way than path a since the latter would generate groups with more separated elements. A further group protection strategy can be seen in path c, in which groups of $k > 4$ will be generated more efficiently than with path a, thus improving the microaggregation outcomes. Note that different protection strategies may exhibit different outcomes according to each dataset and its inherent data distribution.

5. Experiments

In this section, we provide different experiments to showcase the efficacy of our approach by testing it on three datasets that serve as benchmarks. More concretely, we first analyse different aspects related to the computational time of our method in Section 5.1. Next, by using the benchmark datasets, we analyse the IL (expressed in per-cent), to evaluate data utility (*cf.*, Section 2 for details). Note that given a privacy value k that guarantees that k-anonymity is achieved for the microaggregated dataset, low values of IL result in improving the result of microaggregation. Therefore, we compare our proposal to current state-of-the-art methods, and the results of all these experiments are outlined in Section 5.2. Finally, we analyse the trade-off between IL and computational time of our method in Section 5.3.

We used three datasets as benchmarks for our experiments, namely "Census", "EIA" and "Tarragona". These SDC microdata sets are considered *de facto* benchmarks in research on microaggregation (Domingo-Ferrer & Mateo-Sanz, 2002; Templ, 2008). The **Census** dataset was obtained through the *Data Extraction System of the U.S. Bureau of the Census*. It consists of 1080 rows with 13 attributes. The **Tarragona** dataset was obtained from the databases of the Chamber of Commerce of Tarragona. It contains data on 834 companies in the Tarragona area with 13 attributes per record. The **EIA** dataset was obtained from the U.S. Energy Information Authority, and includes 4092 records with 15 variables. Please refer to Templ (2008) for further details about these benchmark datasets.

In the implementation, we selected MDAV due to its efficiency (i.e. the complexity of MDAV is quadratic with respect to the number of records in the dataset (Domingo-Ferrer & Mateo-Sanz, 2002)). For solving the TSP we used the Concorde approach (Applegate et al., 2006), which is currently one of the best approaches.

For the sake of brevity, we refer to the application of TSP without the compression/decompression phases (hence, performing the TSP over the original, r-records data set) as C-MHM. Accordingly, the application of our compression proposal, i.e. executing compression/decompression phases of Algorithm 1, is labelled as Cc-C-MHM, where c denotes the compression factor.

For comparison purposes, we have used MDAV (Domingo-Ferrer & Mateo-Sanz, 2002) and V-MDAV (Solanas & Martínez-Ballesté, 2006).

In the experiments related to running time, we have used an implementation in R with RStudio IDE version 1.3.1093 and the packages TSP version 1.1 and sdcMicro version 5.5.1, running on a computer with 4 × 2.2 GHz Intel Core i7 CPU and 16 GB of RAM. For the sake of reproducibility of the experiments, the code and implementations are publicly available in GitHub.[1]

5.1. Computational time and data distribution

As defined in Morrison, Jacobson, Sauppe, and Sewell (2016), the computational cost of Concorde is $O(Mb^d)$, where M is a limit on the time to explore subproblems and is tied to the number of nodes, d is the search depth and b is a branching factor. To study the impact of a dataset's number of records and the distribution of its elements on the Concorde method, we created two sets of datasets in $\mathbb{R}^2$ with $[100, 500, 1000, 5000, 10000]$ records, namely normal set and uniform set, where:

- In the normal set, the *rnorm()* function from *R-Project* was used, which generates random elements with a normal distribution $N(0, 1)$.

[1] https://github.com/armandomayalopez/Compress-TSP-MHM.

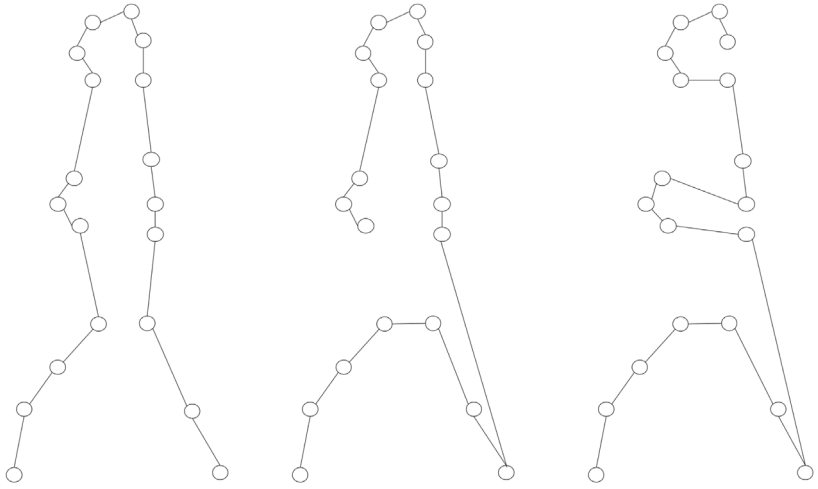


Fig. 3. Example of different grouping protection strategies. Note that paths b and c may generate better microaggregation strategies despite describing longer paths than path a.

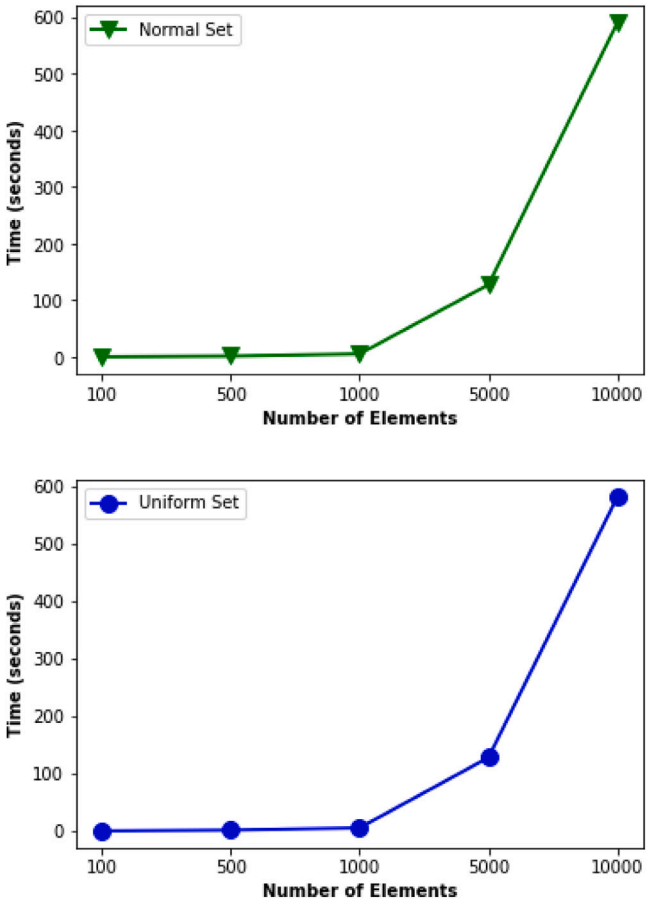


Fig. 4. Time (in seconds) required by our method to create a TSP path according to different data distributions.

- In the uniform set, the *runif()* function from *R-Project* was used, which generates random elements with a uniform distribution $U(0, 1)$.

Next, we applied the C-MHM method to each dataset and depicted the corresponding times in Fig. 4. As observed, there is a relationship between the number of elements of a dataset and the computational time of the method, which follows an exponential growth path. Moreover, note there are no distinguishable differences between both data distributions (cf. Fig. 4). Notably, in datasets with a low number of elements, the difference is almost negligible since most of the time corresponds to system overhead.

Table 1
Computational times (in seconds) of each combination of compression parameter and benchmark.

	Methods				
Dataset	C-MHM	C2-C-MHM	C3-C-MHM	C4-C-MHM	C5-C-MHM
Census	6.58	1.81	0.9	0.56	0.39
Tarragona	4.75	1.71	0.89	0.6	0.44
EIA	75.26	18.28	8.05	4.97	3.19

The next experiment analyses the efficiency due to compression/decompression compared to the original Concorde-MHM method. Table 1 shows the time required to compute the TSP path according to each strategy and benchmark dataset. As observed, the time required by the original Concorde approach is always higher than the one required when including our compression proposal. Moreover, the higher the value of c, the lower the time, which decreases exponentially as seen in Section 'Computational Time and Data Distribution'.

5.2. Microaggregation results

Compression aims to reduce the computational time required by the TSP methods to compute a path while minimising the IL. In this regard, the outcomes of the experiments denote that applying the compression succeeds at fulfilling that objective with more or less efficacy according to their input parameter and the characteristics of the dataset under evaluation. In all cases, compression increases the path length compared to the original Concorde approach. However, they achieve lower, and thus better IL outcomes in some cases, following the observations described in Section 4.2. In the case of Census (cf Table 2) we can observe that the C2-C-MHM strategy obtains similar values as C-MHM, especially for $k > 3$, and outperforms it for $k > 5$. Similarly, the rest of the compression ratios obtain better values as k increases. C5-C-MHM is the strategy that obtains the best outcomes for $k = 10$, despite the notable increase in path length.

The outcomes obtained in the case of Tarragona are depicted in Table 3. The behaviour of applying compression in Tarragona is not as efficient as in Census in terms of IL. For $k > 5$, we achieve values

Table 2
Percentage of IL from the tests using Census. Highlighted values denote the best outcome for each k.

	Census							
Method	Path Length	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
MDAV	N/A	5.6922	7.4947	9.0884	10.3847	11.6688	12.3916	13.3368	14.1559
V-MDAV	N/A	5.6619	7.4947	9.0070	10.2666	11.5999	12.2985	13.3368	14.0730
C-MHM	1173.23	5.0321	6.9691	8.4681	9.7646	11.0744	12.4255	13.7720	14.9954
C2-C-MHM	1308.05	5.2541	6.9869	8.5187	9.4894	10.7869	11.9158	12.8477	13.8309
C3-C-MHM	1531.64	5.6821	7.8676	8.9504	9.5073	11.1161	12.118	12.7336	13.7258
C4-C-MHM	1607.55	6.9922	7.4893	9.7413	10.8816	11.7034	12.096	13.4572	14.1274
C5-C-MHM	1689.13	7.8868	8.8086	9.0884	11.4911	12.2091	12.9724	13.4921	13.6916

Table 3
Percentage of IL from the tests using Tarragona. Highlighted values denote the best outcome for each k.

	Tarragona							
Method	Path Length	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
MDAV	N/A	16.9326	19.5460	22.4619	26.3252	27.5184	29.6929	31.2146	33.1929
V-MDAV	N/A	16.6603	19.5460	22.4619	26.3252	27.5184	29.6929	31.2146	33.1929
C-MHM	772.62	14.8835	18.0649	21.6954	25.0690	27.6827	29.5296	30.7770	32.3488
C2-C-MHM	885.02	15.7625	18.1403	22.5123	25.5584	28.1538	29.4089	31.9773	33.7297
C3-C-MHM	990.96	16.9326	19.6913	22.2274	25.8644	29.1273	30.8186	31.5190	32.3006
C4-C-MHM	1020.69	18.7037	19.5460	23.7335	26.8053	29.9042	30.1385	31.7141	32.7920
C5-C-MHM	1068.95	20.8067	22.3369	22.4616	28.1803	30.9707	32.2069	32.7667	33.0026

Table 4
Percentage of IL from the tests using EIA. Highlighted values denote the best outcome for each k.

	EIA							
Method	Path Length	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
MDAV	N/A	0.4829	0.6713	1.6667	1.3078	2.2141	2.9910	3.4086	3.5474
V-MDAV	N/A	0.4829	0.6713	1.2771	1.2320	2.2038	2.9193	3.2835	2.7478
C-MHM	740.69	0.3704	0.5166	0.7606	1.0554	1.6652	1.8448	1.9348	2.1129
C2-C-MHM	928.97	0.4159	0.6309	0.9423	1.1263	1.6821	1.8587	2.0378	2.1869
C3-C-MHM	992.1	0.4645	0.8101	0.9318	1.0674	1.6948	1.8359	2.0205	2.1734
C4-C-MHM	1049.04	0.5554	0.6443	1.1502	1.2983	1.7335	1.8831	2.0982	2.1756
C5-C-MHM	1271.74	1.3066	1.4934	1.6076	2.2107	2.4945	2.7590	2.9181	2.9567

close to these obtained by the C-MHM method, outperforming them in the case of $k = 8$ with C2-C-MHM, and C3-C-MHM with $k = 10$.

Finally, Table 4 shows the outcomes obtained in the case of EIA. Again, the outcomes between C-MHM and the application of compression are closer the higher the value of k is, especially for $k > 6$. In the particular case of $k = 8$, C3-C-MHM outperforms the original C-MHM. It can also be noticed that in the case of C5-C-MHM, the IL obtained is remarkably worse, indicating that such compression is hindering the inherent group distribution of the dataset. More concretely, due to the particular data distribution of EIA's records, we can observe that some values of k enforce the creation of groups that break such natural disposition. The latter, however, can be remarkably overcome by using variable-sized heuristics.

5.3. Trade-off analysis of TSP-based methods

To better illustrate the efficacy of our proposal in the context of TSP-based heuristics, we created an additional experiment analysing the trade-off between the IL and the computational time, as we aimed to enhance both. Fig. 5 shows, for each dataset, the trade-off analysis of each TSP-based method. Overall, all the outcomes remain close in the x-axis (i.e., denoting a similar range of IL values, with a clear exception in the case of C5-C-MHM and EIA). In contrast, all compression strategies require less time to be computed. In the case of Census, we can observe that the outcomes of C2-C-MHM and C3-C-MHM are almost aligned in the x-axis with these obtained by the C-MHM method (i.e., the circular markers denoting the values for each value of k are almost vertically aligned). Moreover, applying compression we always obtain lower IL values the higher the value of k, which is particularly obvious for $k = 10$. In the latter case, the original C-MHM obtains a value that is isolated from the rest, both in terms of IL and time. The outcomes of Tarragona resemble those obtained by Census, yet in this case, only C2-C-MHM seems to obtain similar values to C-MHM in terms of IL. However, as seen in Census, these values get closer the higher the value of k is. Finally, the outcomes of the EIA dataset show that almost all strategies but C5-C-MHM obtain similar values to those achieved by C-MHM, yet with much more efficiency.

In summary, these outcomes justify the applicability of our compression method both in small and big datasets, despite being the latter the ones that reflect the trade-off gain in a more evident manner. However, different datasets may exhibit different behaviours and thus require careful analysis to select the most appropriate strategy according to the trade-off optimisation criteria.

6. Discussion

Finding the optimal microaggregation is NP-Hard. Hence, researchers have devoted extensive efforts to finding good but sub-optimal solutions for decades (Zigomitros et al., 2020). Therefore, it is not straightforward to find fresh solutions that advance the state of

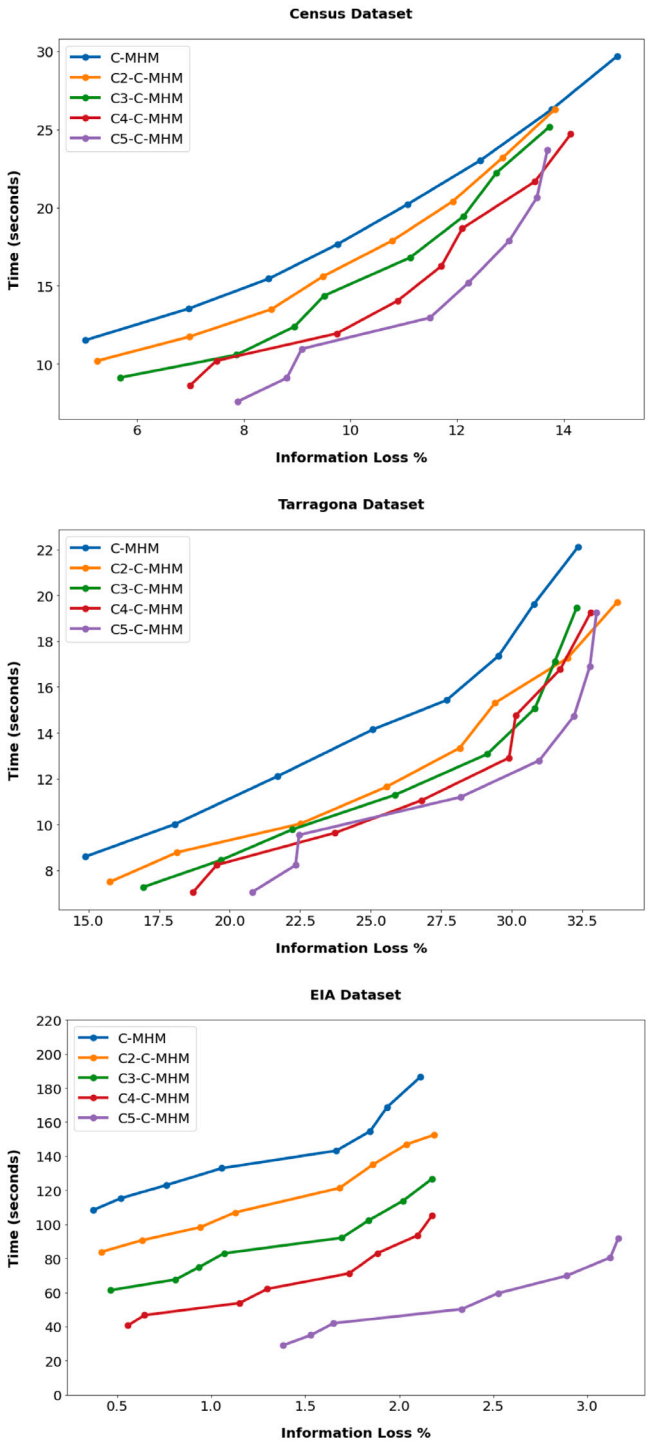


Fig. 5. Trade-off between IL and computational time of the TSP-based methods. The circular markers of the series denote the values obtained from $k = 3$ to $k = 10$ (i.e., left to right).

the art, let alone if we aim to improve both the IL and the efficiency of such solutions.

Despite providing the lowest IL values, TSP-based microaggregation methods suffer from scalability issues (Maya-López et al., 2021). Therefore, developing strategies to reduce the computational time of TSP heuristics is crucial. Going a step beyond, we aimed to develop a strategy that can be applied regardless of the dataset's size and improve the outcomes provided by classical TSP approaches.

As observed in Section 5.1, all compression strategies succeed in reducing the computational time of C-MHM. Moreover, as discussed in Section 5.2, and especially for high values of k, compression strategies enhance the IL values, which sometimes are close or even better than the ones achieved by the original C-MHM. In this regard, different compression values are combined with different k to study the impact of different configurations according to different types of datasets. Overall, as seen in Section 5.3, the trade-off between IL and computational time is always beneficial when applying Compress strategies, especially those with lower parameter values. Moreover, the larger the number of dataset records, the more significant the compression's impact on reducing the computational time.

Therefore, our approach enables the application of different configurations to satisfy an optimisation criteria (i.e., balancing IL and computational time according to each application context). Moreover, contrary to other methods (e.g. Monedero et al., 2019) our approach is applicable without data constraints (i.e. it can be applied regardless of the nature of the data and its size, achieving remarkable outcomes even in small datasets), and improves both the utility of the data and the microaggregation performance.

Finally, some limitations of our work are worth noting. For instance, the capability of our method to improve both the IL and the computational time depends on the particularities of each dataset. Another slight drawback of our method is that it requires a structure storing the relationship between the records of the original dataset and the compressed dataset to reverse such compression in the last step. However, since such a structure is one-dimensional, it does not hinder the scalability of our approach.

7. Conclusion

Microaggregation heuristics have been extensively explored in the past, often with the aim to improve their data utility.

After an extensive analysis of the state of the art, we noticed that the research area targeted in our article is not receiving a lot of attention from researchers. Therefore, considering the advent of Big Data and more accurate yet costly methods such as TSP, developing heuristics aiming to reduce the computational time of privacy protection heuristics such as the ones based on microaggregation is crucial.

In this article, we proposed an efficient Compression-based TSP heuristic for microaggregation, which outperforms the state of the art in terms of trade-off between IL and computational time, according to extensive experiments and comparisons. Moreover, in some cases, our Compression heuristic is able to preserve the natural distribution of data more efficiently than the original method and thus, generate groups that increase the utility while reducing the computational time.

Part of our future work includes the extension of the Compress method to other areas and applications further than microaggregation, where a trade-off between data utility and performance is assumable. Moreover, we plan to analyse other group protection heuristics, including variable-sized clustering, to enhance the utility of the protected data.

CRediT authorship contribution statement

Armando Maya-López: Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Antoni Martínez-Ballesté:** Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision. **Fran Casino:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the project *LOCARD* (https://locard.eu) (Grant Agreement no. 832735), by the Spanish Ministry of Science & Innovation with project IoTrain RTI2018-095499-B-C32, by the Government of Catalonia, Spain with grant 2017-DI-002, and projects 2017-SGR-896 and ACTUA 2020PANDE00103, and by Universitat Rovira i Virgili, Spain with project 2017PFR-URV-B2-41. F. Casino was supported by the Beatriu de Pinós programme of the Government of Catalonia (Grant No. 2020 BP 00035). The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

References

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). *The traveling salesman problem: A computational study*. Princeton University Press.

Casino, F., Domingo-Ferrer, J., Patsakis, C., Puig, D., & Solanas, A. (2015). A k-anonymous approach to privacy preserving collaborative filtering. *Journal of Computer and System Sciences*, *81*, 1000–1011.

Casino, F., Patsakis, C., & Solanas, A. (2019). Privacy-preserving collaborative filtering: A new approach based on variable-group-size microaggregation. *Electronic Commerce Research and Applications*, *38*, Article 100895.

Chang, C. -C., Li, Y. -C., & Huang, W. -H. (2007). TFRP: An efficient microaggregation algorithm for statistical disclosure control. *Journal of Systems and Software*, *80*, 1866–1878.

Climer, S., Zhang, W., & Joachims, T. (2006). Rearrangement clustering: Pitfalls, remedies, and applications. *Journal of Machine Learning Research*, *7*.

Domingo-Ferrer, J., Martinez-Ballesté, A., Mateo-Sanz, J. M., & Sebé, F. (2006). Efficient multivariate data-oriented microaggregation. *The VLDB Journal*, *15*, 355–369.

Domingo-Ferrer, J., & Mateo-Sanz, J. M. (2002). Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, *14*, 189–201.

Domingo-Ferrer, J., Sebé, F., & Solanas, A. (2008). A polynomial-time approximation to optimal multivariate microaggregation. *Computers & Mathematics with Applications*, *55*, 714–732. http://dx.doi.org/10.1016/j.camwa.2007.04.034.

European Commission (2018). 2018 Reform of EU data protection rules. URL: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf.

Fayyoumi, E., & Oommen, B. J. (2010). A survey on statistical disclosure control and micro-aggregation techniques for secure statistical databases. *Software - Practice and Experience*, *40*, 1161–1188.

Hansen, S. L., & Mukherjee, S. (2003). A polynomial algorithm for optimal univariate microaggregation. *IEEE Transactions on Knowledge and Data Engineering*, *15*, 1043–1044.

Heaton, B., & Mukherjee, S. (2011). Record ordering heuristics for disclosure control through microaggregation. In *Proceedings of the international conference on advances in communication and information technology*.

Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on theory of computing*. (pp. 604–613).

Laszlo, M., & Mukherjee, S. (2005). Minimum spanning tree partitioning algorithm for microaggregation. *IEEE Transactions on Knowledge and Data Engineering*, *17*, 902–911.

Liew, C. S., Abbas, A., Jayaraman, P. P., Wah, T. Y., Khan, S. U., et al. (2016). Big data reduction methods: A survey. *Data Science and Engineering*, *1*, 265–284.

Lin, J. -L., Wen, T. -H., Hsieh, J. -C., & Chang, P. -C. (2010). Density-based microaggregation for statistical disclosure control. *Expert Systems with Applications*, *37*, 3256–3263.

Maya-López, A., Casino, F., & Solanas, A. (2021). Improving multivariate microaggregation through Hamiltonian paths and optimal univariate microaggregation. *Symmetry*, *13*, http://dx.doi.org/10.3390/sym13060916, URL: https://www.mdpi.com/2073-8994/13/6/916.

Monedero, D. R., Mezher, A. M., Colomé, X. C., Forné, J., & Soriano, M. (2019). Efficient k-anonymous microaggregation of multivariate numerical data via principal component analysis. *Information Sciences*, *503*, 417–443.

Morrison, D. R., Jacobson, S. H., Sauppe, J. J., & Sewell, E. C. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, *19*, 79–102. http://dx.doi.org/10.1016/j.disopt.2016.01.005, URL: https://www.sciencedirect.com/science/article/pii/S1572528616000062.

Mortazavi, R., & Jalili, S. (2014). Fast data-oriented microaggregation algorithm for large numerical datasets. *Knowledge-Based Systems*, *67*, 195–205. http://dx.doi.org/10.1016/j.knosys.2014.05.011, URL: http://www.sciencedirect.com/science/article/pii/S0950705114001956.

Mortazavi, R., Jalili, S., & Gohargazi, H. (2013). Multivariate microaggregation by iterative optimization. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *39*, 529–544.

Osaba, E., Yang, X. -S., & Del Ser, J. (2020). Traveling salesman problem: A perspective review of recent research and new results with bio-inspired metaheuristics. *Nature-Inspired Computation and Swarm Intelligence*, 135–164.

Panagiotakis, C., & Tziritas, G. (2011). Successive group selection for microaggregation. *IEEE Transactions on Knowledge and Data Engineering*, *25*, 1191–1195.

Samarati, P., & Sweeney, L. (1998). *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression: Technical report*, SRI International.

Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y., & Herawan, T. (2014). Big data clustering: A review. In *International conference on computational science and its applications* (pp. 707–720). Springer.

Shmoys, D. B., Lenstra, J., Kan, A. R., & Lawler, E. L. (1985). *The traveling salesman problem, volume 12*. John Wiley & Sons, Incorporated.

Solanas, A., González-Nicolás, U., & Martínez-Ballesté, A. (2010). A variable-MDAV-based partitioning strategy to continuous multivariate microaggregation with genetic algorithms. In *The 2010 international joint conference on neural networks* (pp. 1–7). http://dx.doi.org/10.1109/IJCNN.2010.5596660.

Solanas, A., & Martínez-Ballesté, A. (2006). VMDAV: A multivariate microaggregation with variable group size. In *17th COMPSTAT symposium of the IASC, Rome* (pp. 917–925).

Solé, M., Muntés-Mulero, V., & Nin, J. (2012). Efficient microaggregation techniques for large numerical data volumes. *International Journal of Information Security*, *11*, 253–267.

Templ, M. (2008). Statistical disclosure control for microdata using the R-package sdcmicro. *Transactions on Data Privacy*, *1*, 67–85.

Willenborg, L., & de Waal, T. (2001). *Elements of statistical disclosure control*. Springer New York, http://dx.doi.org/10.1007/978-1-4613-0121-9.

Yang, G., Ye, X., Fang, X., Wu, R., & Wang, L. (2020). Associated attribute-aware differentially private data publishing via microaggregation. *IEEE Access*, *8*, 79158–79168.

Zaidi, T. (2020). Travelling salesman problem and its applications. *International Journal of Mathematics, Game Theory, and Algebra*, *29*, 73–80.

Zigomitros, A., Casino, F., Solanas, A., & Patsakis, C. (2020). A survey on privacy properties for data publishing of relational data. *IEEE Access*, *8*, 51071–51099. http://dx.doi.org/10.1109/ACCESS.2020.2980235.