Consiglio Nazionale delle Ricerche
Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti"

C. Gentile, E. Spagnolo-Arrizabalaga, J. Castro

AN ALGORITHM FOR THE MICROAGGREGATION PROBLEM USING COLUMN GENERATION

R. 20-02, 2020

Claudio Gentile – Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche, Rome (Italy). Email: gentile@iasi.cnr.it.

Enric Spagnolo-Arrizabalaga – DerbySoft, Avinguda Diagonal, 472, Esc A, 8º 3ª, 08006, Barcelona, Spain. Email: enric.spagnolo@derbysoft.net.

Jordi Castro – Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona. Email: jordi.castro@upc.edu.

ISSN: 1128–3378

ISSN: 1128-3378 – "Collana dei rapporti dell'Istituto di Analisi dei Sistemi ed Informatica"

Research Report Series of

Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" – CNR

via dei Taurini 19, 00185 ROMA, Italy

tel. ++39-06-4993-7102/1
fax ++39-06-4993-7106
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

The field of Statistical Disclosure Control aims at reducing the risk of re-identification of an individual when disseminating data, and it is one of the main concerns of national statistical agencies. Operations Research (OR) techniques were widely used in the past for the protection of tabular data, but not for microdata (i.e., files of individuals and attributes). This work presents (as far as we know, for the first time) an application of OR techniques for the microaggregation problem, which is considered one the best methods for microdata protection and it is known to be NP-hard.

The new heuristic approach is based on a column generation scheme and, unlike previous (primal) heuristics for microaggregation, it also provides a lower bound on the optimal microaggregation. Computational results on real data typically used in the literature show that solutions with small gaps are often achieved and that dramatic improvements are obtained with respect to the most popular heuristics in the literature.

Keywords Integer Programming, Column Generation, Data Privacy, Clustering, Microaggregation

1. Introduction

Several private corporations and public agencies (among them, national statistical agencies) store information about individuals (either companies or persons) in the form of microdata files. A microdata file of n individuals and t variables (or attributes) is, in practice, a $n \times t$ matrix whose element (i, j) provides the value of attribute j for individual i. Formally, it can be defined as a mapping

$$V : N \subseteq P \rightarrow D(V_1) \times D(V_2) \times \ldots \times D(V_t),$$

where P is a population, N is a subset of the population, called sample, and $D(V_i)$ is the domain of the attribute $i \in \{1, \ldots, t\}$. Depending on the domain, variables can be classified as numerical (e.g., "age", "gross domestic product") or categorical (e.g, "gender", "country"). Crossing one or several categorical variables gives rise to a table of data, which is formally defined as the mapping

$$T : D(V_{i_1}) \times D(V_{i_2}) \times \cdots \times D(V_{i_l}) \rightarrow \mathbb{R} \text{ or } \mathbb{N},$$

l being the number of categorical variables that were crossed. The result of function T (cell values) belongs to $\mathbb{N}$ for a frequency table (e.g., number of persons per gender and country), and to $\mathbb{R}$ for a magnitude table (e.g., salary of persons per gender and country).

Microdata and tabular data are the two most usual types of data disseminated by national statistical agencies. In either form, data has to be treated before publication in order to prevent the re-identification (by external users named *attackers*) of confidential information of individuals (e.g., "salary"). Attackers usually consider groups of attributes (named *almost identifiers*) that can be used to identify an individual. The set of methods for microdata or tabular data protection are encompassed in the so called *Statistical Disclosure Control* (SDC) topic [19, 6]. SDC methods attempt to reduce the disclosure or re-identification risk [1],[19, Chapter 3].

Broadly, data protection methods (for either microdata or tabular data) can be classified in *perturbative* (i.e., they modify the original data) and *non-perturbative* (they do not modify the data, instead, they suppress part of them or modify their structure). Operations Research (OR) techniques have been widely used in the past for tabular data protection, either in perturbative methods (such as controlled tabular adjustment [3, 8, 7, 16, 18], and controlled rounding [27, 25]) or non-perturbative ones (such as the cell suppression problem [2, 5, 14, 13, 26]). For microdata, among the most relevant perturbative methods we find microaggregation [10, 12, 29], rank swapping [9, 22], and data shuffling [23]; non-perturbative methods for microdata are related with the recoding of the categorical variables [19, Chapter 3]. However, unlike the case of tabular data, OR tools (as far as we know) have not been applied for microdata, even when some of those methods formulate and need to solve a combinatorial optimization problem. This is the case of the microaggregation problem, which is the object of this work. It is worth to note that in the empirical comparison made in [11], using several scores, rank swapping and microaggregation were the two more performant methods for microdata in terms of trade-off between disclosure risk and information loss. All techniques except microaggregation are out of the scope of this paper.

In brief, microaggregation aims at grouping points in clusters of a minimum size k, which is a parameter of the problem, replacing the original data by the centroids of the clusters. It is worth noting that microaggregation, which is known to be NP-hard [24], is significantly different from other classical clustering methods, such as k-medians or k-means [15] (which are related to the p-median problem in facility location [21]). In microaggregation, the parameter k fixes the minimum number of points per cluster, while the number of clusters is free; on the other hand, k in k-medians or k-means fixes the number of clusters, with no constraint on the cardinality of each cluster.

The structure of the rest of the paper is as follows. Section 2 defines and provides some background on the microaggregation problem. Sections 3 and 4 introduce, respectively, a novel integer programming model, and a column generation approach for the microaggregation problem. The final algorithm implemented and the computational results are presented in Sections 5 and 6.

2. Microaggregation

Microaggregation is a *perturbative* technique mainly considered for numeric variables which arises from the concept of *k-anonymity* [28]:

Definition 2.1. *Given $k \in \mathbb{N}$, $k \geq 2$, let V be a microdata with n individuals $n \geq k$ and t attributes $V_1, \ldots, V_t$. Let $g = (V_{j_1}, \ldots, V_{j_m})$ be an almost identifier for V, $j_q \in \{1, \ldots, t\}$, $q = 1, \ldots, m$. Then we say V is k-anonymous if for every possible value in $D(V_{j_1}) \times \ldots \times D(V_{j_m})$, there exist either 0 or at least k individuals in V with this value for the attributes in g.*

Microaggregation intends to modify the values for the attributes involved in a given almost identifier so that eventually the microdata satisfies *k-anonymity* for this considered almost identifier. Therefore, it first joins different individuals of the microdata file in sets of at least k individuals. Then, for each of these sets of individuals, it substitutes the values of the attributes of the given almost identifier by common values for all the individuals in the set. This way *k-anonymity* for the given almost identifier is satisfied by construction in the modified microdata. The resulting sets of individuals will be called *clusters* from now on. A partition of the whole set of individuals into clusters is called a *clustering*. We note that other clustering techniques such as k-medians or k-means cannot be used for k-anonimity since they not constraint the size of clusters.

Generally the common values taken for a cluster (after data perturbation) are the centroid of the cluster, that is a record of values which reduces as much as possible the information loss, or *spread*, after the aggregation. With this idea it is clear that the objective of microaggregation technique is to minimize the total sum of distances of the data of the individuals to the centroids of their respective clusters. In practical cases the value of k is relatively small (classical microaggregation uses k around 3, see [10]).

Example 1. *Let $g = (Employees, Surface)$ be a numeric almost identifier for a microdata of industrial factories. Suppose we want to achieve k-anonymity with $k = 2$ and that our microaggregation procedure suggests us to join the 3 factories in Table 1 to form a cluster. The centroid of this group (average for the values of the attributes of the almost identifier) is $\frac{55+48+41}{3} = 48$ employees and $\frac{1410+1205+1120}{3} = 1245$ m^2 of surface. Therefore in the eventual published microdata the factories f_1, f_2 and f_3 will all have 48 employees and 1245 m^2 of surface.*

We introduce a measure of the spread and therefore we define the problem in terms of the data to aggregate. For simplicity, from now on we consider only attributes related to an almost identifier. We make an abuse of notation when referring to the square of a vector v as $v^2 = v^T v$, for $v \in \mathbb{R}^m$. We can restate our problem in the following way. Suppose we have n individuals data vectors a_i, $i \in \{1, \ldots, n\}$, with the attributes of the almost identifier. Let $k \geq 2$ be the integer that fixes the anonymity. The target is to partition the individuals a_i into q clusters with size $n_s \geq k$, for all $s \in \{1, \ldots, q\}$, to satisfy k-anonymity, such that:

$$\sum_{s=1}^{q} \sum_{j=1}^{n_s} d(a_{s_j}, \overline{a_s})$$

is minimized, where $d(a, b)$ is a distance between data vectors a and b, q is the number of clusters, a_{s_j} is the element j in cluster s and $\overline{a_s}$ is the centroid of cluster s, i.e.:

$$\overline{a_s} = \frac{1}{n_s} \sum_{j=1}^{n_s} a_{s_j} \tag{1}$$

Factory	Employees	Surface (m^2)
f_1	55	1410
f_2	48	1205
f_3	41	1120

Table 1: Values for the attributes in g in the original microdata.

The function introduced above is a general measure of the information loss, or spread, in microaggregation. In general the distance $d(\cdot,\cdot)$ considered is the Euclidean distance. For practical reasons, we minimize the sum of its square values. This term is usually named *Sum of Squares Error* (SSE) [10]:

$$SSE = \sum_{s=1}^{q}\sum_{j=1}^{n_s}(a_{s_j} - \overline{a_s})^T(a_{s_j} - \overline{a_s}) = \sum_{s=1}^{q}\sum_{j=1}^{n_s}(a_{s_j} - \overline{a_s})^2 \tag{2}$$

From now on, we will denote as *feasible clustering* a partition into clusters of size at least k for each of them. The following Proposition gives an upper bound on the cardinality of each cluster.

Proposition 2.2. *[10] Any cluster belonging to an optimal microaggregation must have size less than or equal to* $2k-1$.

The proof is very simple and it simply relies on the following fact. Given a feasible clustering π containing a cluster with size greater than or equal to $2k$, we can split this cluster into two clusters of size greater than or equal to k obtaining a clustering π'. It is clear that the sum of the distances of the members of the two clusters from the new two centroids is reduced and therefore $SSE_{\pi'} \leq SSE_{\pi}$.

We distinguish the cases of univariate data and multivariate data. *Univariate data* occurs when the vectors a_i have one single attribute and *multivariate data* when there are more than one attribute. For the particular case of univariate data, optimal microaggregation can be achieved with a polynomial algorithm based on shortest paths in a weighted graph [17].

In the case of multivariate data, optimal microaggregation is an *NP-hard* problem, that is it should not be expected to be solved in polynomial time. There exists a lot of literature about heuristic algorithms that obtain feasible clusterings with reasonable SSE. We cite the most remarkable ones: (i) *Maximum Distance (MD)* [10]; (ii) *Maximum Distance to Average Value (MDAV)* [12]; (iii) *Variable-Maximum Distance to Average Value (V-MDAV)* [29]. Experimental results discussed in [29] show that V-MDAV outperforms the two previous ones in synthetic grouped data and also that MDAV and MD have similar performance in general although MDAV has a lower computational cost.

3. IP Models for Microaggregation

In this section we model Microaggregation with binary variables. We give an exact Integer Programming (IP) formulation for the problem. Unfortunately, it results to be nonlinear, therefore we also present a new Integer Linear Programming (ILP) formulation based on Column Generation. We need the following preliminary result:

Proposition 3.1. *Given a cluster* C_s *containing* n_s *elements denoted by* $\{a_{s_j} \mid j \in \{1,\ldots,n_s\}\}$, *let* $\overline{a_s}$ *be its centroid as defined in (1). Then,*

$$n_s\sum_{j=1}^{n_s}(a_{s_j} - \overline{a_s})^2 = \frac{1}{2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}(a_{s_i} - a_{s_j})^2. \tag{3}$$

Proof. Developing the square product and substituting $\overline{a_s}$ the left hand side equals to:

$$n_s\sum_{j=1}^{n_s}a_{s_j}^2 + n_s\sum_{j=1}^{n_s}\frac{1}{n_s^2}\left(\sum_{i=1}^{n_s}a_{s_i}\right)^2 - 2n_s\sum_{j=1}^{n_s}\left(a_{s_j}\frac{1}{n_s}\sum_{i=1}^{n_s}a_{s_i}\right)$$

In particular, the term

$$n_s\sum_{j=1}^{n_s}\frac{1}{n_s^2}\left(\sum_{i=1}^{n_s}a_{s_i}\right)^2 = \frac{1}{n_s}\sum_{j=1}^{n_s}\left(\sum_{i=1}^{n_s}a_{s_i}\right)^2 = \left(\sum_{i=1}^{n_s}a_{s_i}\right)^2$$

Finally substituting $\big(\sum_{i=1}^{n_s} a_{s_i}\big)^2 = \sum_{j=1}^{n_s}\sum_{i=1}^{n_s} a_{s_j}^T a_{s_i}$ we obtain that the left hand side of (3) is equal to:

$$n_s \sum_{j=1}^{n_s} a_{s_j}^2 - \sum_{j=1}^{n_s}\sum_{i=1}^{n_s} a_{s_j}^T a_{s_i}. \tag{4}$$

Then, in the right hand side of (3), simply substitute the square product and using the following relations,

$$\frac{1}{2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} a_{s_i}^2 = \frac{n_s}{2}\sum_{i=1}^{n_s} a_{s_i}^2 \quad \text{and} \quad \frac{1}{2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} a_{s_j}^2 = \frac{n_s}{2}\sum_{j=1}^{n_s} a_{s_j}^2,$$

the right hand side in (3) is reduced exactly to (4). ∎

This result immediately means that the contribution of cluster C_s to the total SSE (2) is,

$$\sum_{j=1}^{n_s}(a_{s_j} - \overline{a_s})^2 = \frac{1}{2n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}(a_{s_i} - a_{s_j})^2. \tag{5}$$

Let N be the set of microdata individuals. Let us define the binary variables z_{ij} for all pairs $i, j \in N \times N, i \neq j$:

$$z_{ij} = 1 \Leftrightarrow a_i, a_j \text{ belong to the same cluster after microaggregation}$$
$$z_{ij} = 0 \qquad \text{otherwise.}$$

To simplify the writing we will denote with i the corresponding individual a_i and with n_i be the number of nodes of the cluster where a_i belongs to. Using relation (5) we can easily derive that

$$SSE = \frac{1}{2}\sum_{i=1}^{n} \frac{\sum_{\substack{j=1 \\ j\neq i}}^{n} (a_i - a_j)^2 z_{ij}}{n_i}$$

Now we express n_i in terms of the variables z_{ij}:

$$n_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} z_{ji} + 1$$

And finally, substituting the term in the expression of SSE, we obtain an expression of the total SSE in terms only of the variables z_{ij}, $i \neq j$:

$$SSE = \frac{1}{2}\sum_{i=1}^{n} \frac{\sum_{j=1,\ j\neq i}^{n} (a_i - a_j)^2 z_{ij}}{\sum_{j=1, j\neq i}^{n} z_{ij} + 1}. \tag{6}$$

Our initial objective was to construct the clusters such that SSE was minimized. Consequently we should obtain the values of the variables z_{ij} such that they define a feasible clustering and they minimize the expression for SSE in equation (6). Forcing them to define a feasible clustering will give us the constraints. Equation (6) will define our objective function to minimize.

To force the variables to describe clusters, we must express two conditions. First, clusters must be *complete* in the sense that every node in the cluster is related to every other node in the same cluster. For example if i and r are in the same cluster (i.e., $z_{ir} = 1$) and j and r are on the same cluster too (i.e., $z_{jr} = 1$) then it must be that i and j are in the same cluster ($z_{ij} = 1$). It can be described by the following set of inequalities, named *triangle inequalities*:

$$z_{ir} + z_{jr} - z_{ij} \leq 1 \ \text{ for all } i, j, r \in N, \quad i \neq j,\ r \neq j,\ i \neq r$$

From now on we will denote as *cliques* the description of clusters in terms of variables z_{ij}. We will refer to the size of a clique as the size of the cluster it corresponds to.

Second, we must force the clusters (or cliques) to have size at least k. This means $n_i \geq k \; \forall \; i \in N$. We will call it the *size inequality* and it can be easily expressed as:

$$\sum_{j=1,\; j\neq i}^{n} z_{ij} \geq k-1$$

Putting all pieces together, our clustering problem can be described as an optimization problem with the binary symmetric variables z_{ij}, $i, j \in N$, $i \neq j$:

$$\begin{array}{ll} \min \frac{1}{2} \sum\limits_{i=1}^{n} \dfrac{\sum\limits_{j=1,\; j\neq i}^{n} (a_i - a_j)^2 z_{ij}}{\sum\limits_{j=1,\; j\neq i}^{n} z_{ij} + 1} & \\ \text{subject to:} & \\ z_{ij} = z_{ji}, & \text{for all } i, j \in N, \quad i \neq j \\ \sum\limits_{j=1, j\neq i}^{n} z_{ij} \geq k-1, & \text{for all } i \in N \\ z_{ir} + z_{rj} - z_{ij} \leq 1, & \text{for all } i, j, r \in N, \quad i < r < j \\ z_{ij} \in \{0,1\}, & \text{for all } i, j \in N, \quad i \neq j \end{array} \tag{7}$$

The most remarkable issues are that:

1. The function is non-linear because of the dividing term in the fraction (size of a cluster).
2. The function is non-convex. For example for the case $n = 2$ there is only one variable $z = z_{12} = z_{21}$. Let $c = (a_1 - a_2)^2 > 0$. The objective function in (7) is:

$$f(z) = \frac{cz}{1+z} \Rightarrow f''(z) = \frac{-2c}{(1+z)^3}$$

The second derivative is clearly non positive for $z > -1$ and therefore also for $z \geq 0$ (note that z_{12} is non-negative).

This means that, using model (7), the problem has serious difficulties to be solved by ILP enumerative schemes, because one should at least define an effective relaxation.

4. A Column Generation approach

In this section, we introduce a Column Generation approach for the Microaggregation problem inspired by [20]. In our case we modify the weight of a cluster coefficient and more importantly the whole Pricing Problem scheme for new cluster generation.

4.1. Master Problem & Column Generation Scheme

Taking into account Proposition 2.2 we define $\mathcal{C}^* = \{C \subseteq N \,|\, k \leq |C| \leq 2k-1\}$ as the set of feasible clusters for the microaggregation. Let x_C be the binary variable that indicates whether cluster $C \in \mathcal{C}^*$ is or not in the solution for microaggregation. Let w_C be the weight of C; from (5) we derive that w_C can be computed as:

$$w_C = \frac{1}{2} \sum_{i \in C} \frac{\sum\limits_{j \in C} (a_i - a_j)^2}{|C|} = \sum_{i,j \in C} \frac{(a_i - a_j)^2}{|C|}. \tag{8}$$

On account of this we can formulate the microaggregation problem as follows:

$$\begin{array}{ll} \min \sum_{C\in\mathcal{C}^*} w_C x_C & \\ \sum_{C\in\mathcal{C}^*: v\in C} x_C = 1 & v \in N \\ x_C \in \{0,1\}, & C \in \mathcal{C}^* \end{array} \tag{9}$$

The Master Problem, that is obtained from (9) by considering a subset $\overline{\mathcal{C}} \subseteq \mathcal{C}^*$, is similar to the one in [20] with only the modification in the weight of a cluster provided by equation (8). When it comes to the Pricing Problem there raise important differences.

In [20], the Pricing Problem looks for a cluster of size at least k that minimizes the sum of edge weights minus the sum of node weights in this cluster. For the Microaggregation formulation (9), the Pricing Problem must minimize the *average* (8) of the edge weights minus the sum of the node weights in the cluster. We solve it by considering k Pricing Subproblems, each one finding a cluster of fixed size η, for $\eta \in \{k, \ldots, 2k-1\}$, that minimizes the sum of the edge weights divided by η and node weights.

$$\bar{w}_C = w_C - \sum_{v\in C} \lambda_v = \sum_{u,v\in C} \frac{(a_u - a_v)^2}{\eta} - \sum_{v\in C} \lambda_v. \tag{10}$$

It is clear that, when none of these k Pricing Subproblems adds a cluster with negative reduced cost, then there are no more feasible clusters to consider in the Master Problem. At this point, the solution of the Master Problem must be checked: if the solution is binary then, the scheme has provided with an optimal microaggregation; else if the solution is fractional we obtained a lower bound for the Microaggregation problem that can be used in a Branch&Price scheme. Note that this is the first approach in the literature for the computation of a lower bound for SSE.

4.2. Pricing Subproblem with fixed cluster size

In order to solve the Pricing Subproblem for fixed cluster size η we can set up an ILP formulation based on a graph representation on the complete undirected graph $G = (N, A)$ with edge weights $(a_i - a_j)^2/\eta$, $e = ij \in A$, and node weights, λ_i, $i \in N$. We look for a feasible cluster $C \in \mathcal{C}^*$, $|C| = \eta$, such that the difference between its edge weights and its node weights is negative.

Previous models for cluster generation (e.g., see [20]) include variables for every node and edge in G. Here we propose a new ILP model considering only edge variables.

First of all, we suppose $k \geq 2$ and therefore also $\eta \geq 2$. Suppose C is the new cluster is going to be generated with the Pricing Subproblem for fixed cluster size η. We define variables z_{ij}, $ij \in A$, such that:

$$\begin{array}{ll} z_{ij} = 1 & i, j \text{ are in cluster } C, \\ z_{ij} = 0 & \text{otherwise.} \end{array}$$

Note that with these variables the solution of the Pricing Subproblem will be a clique of size η only if

$$\sum_{e\in A} z_e = \frac{\eta(\eta-1)}{2}. \tag{11}$$

Equation (11) will be referred as the *fixed size constraint*. Another group of constraints could be the so called triangle inequalities, exactly as in Section 3. Using them, together with nonnegativity constraints $z_{ij} \geq 0$, $ij \in A$, we can ensure the problem solution provides with a clique in edges:

$$z_{ir} + z_{rj} - z_{ij} \leq 1 \;\; i, r, j \in N, i < r < j.$$

The above constraints lack to force the solution to represent a single clique. For example if we consider the case $n = 6$, $N = \{1, 2, 3, 4, 5, 6\}$, $\eta = 3$, the solution provided by activating edges $(1, 2)$, $(3, 4)$ and $(5, 6)$ satisfies the fixed size constraint and the triangle inequalities. It is actually a clique partitioning into three separated subcliques. We must discard this type of solution by forcing connectivity. To do that we introduce the *node-to-node inequalities*.

Definition 4.1. *Let i, $j \in N$, $i \neq j$. We define the* node-to-node inequality *for i and j as,*

$$\sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta - 2) z_{ij} \geq 0 \tag{12}$$

This inequality is clearly valid, indeed, if $z_{ij} = 1$, it simply requires that i is connected to at least $\eta - 2$ nodes different from i and j; otherwise it is implied by nonnegativity constraints. However, if we consider the above example, constraints (12) are clearly violated for instance for the pair $i = 1$ and $j = 2$. Note that the inequality (12) associated with j and i is a different inequality belonging to the same class of constraints.

Next proposition shows that the nonnegativity constraints, the fixed size constraint, and the node-to-node constraints are sufficient to describe the clusters with fixed size.

Proposition 4.2. *Let $\eta \geq 2$, the binary feasible solutions satisfying nonnegativity constraints, the fixed size constraint (11), and the node-to-node constraints (12) represent cliques in G with size η.*

Proof. The case $\eta = 2$ is trivial. For $\eta \geq 3$, let us suppose the edge (v, w) is activated between nodes v, $w \in N$. Imposing the node-to-node inequality,

$$\sum_{e \in \delta(v) \setminus (v,w)} z_e - (\eta - 2) z_{vw} \geq 0 \tag{13}$$

means there are at least $\eta - 2$ other nodes connected to v apart from w. Let us call $\mathcal{K} \subset N$ this set of nodes, $|\mathcal{K}| \geq \eta - 2$. Then we impose the reverse node-to-node inequality for w and v:

$$\sum_{e \in \delta(w) \setminus (v,w)} z_e - (\eta - 2) z_{vw} \geq 0$$

which forces w to be connected to at least $\eta - 2$ nodes apart from v. To start we can suppose first those nodes are $\mathcal{K}$ too. Then we take k_1 a node in $\mathcal{K}$ and we impose the node-to-node inequality for k_1, v. As we are supposing k_1 is connected to both v, w, it means k_1 is connected to at least $\eta - 3$ other nodes. We can suppose those nodes are yet the other nodes in $\mathcal{K}$. We could do the same for the next k_t nodes in $\mathcal{K}$. At every stage we would be adding exactly $|\mathcal{K}| - t$ new edges.
If $|\mathcal{K}| = \eta - 2$, then the resulting amount of edges would be

$$\begin{aligned}
&1 + \eta - 2 + \eta - 2 + \sum_{t=1}^{\eta-2} \eta - 2 - t = \\
&= 2\eta - 3 + (\eta - 2)^2 - \sum_{t=1}^{\eta-2} t = \\
&= 2\eta - 3 + (\eta - 2)^2 - \frac{(\eta - 2)(\eta - 1)}{2} = \\
&= \frac{4\eta - 6 + 2(\eta^2 - 4\eta + 4) - (\eta^2 - 3\eta + 2)}{2} = \frac{\eta(\eta - 1)}{2}
\end{aligned}$$

and the fixed size constraint would be satisfied. Besides, the solution would be a clique. Note that if $|\mathcal{K}| > \eta - 2$, then, applying successively the node-to-node inequality as above, would lead us to a greater amount of edges and this would violate the fixed size constraint. More than that, if at some point our construction was not strictly satisfied, in the sense that at some stage t the nodes connected to k_t where not exactly the nodes in $\mathcal{K}$ plus v and w, then, there would be an external node $z \notin \mathcal{K} \cup \{v, w\}$ connected to k_t. If we took the node-to-node inequality for z, k_t this node z would be connected to at least $\eta - 2$ nodes different from k_t. The associated edges in $\delta(z)$ should be added in the sum of edges described above violating the equality between the total amount of edges and the right-hand-side of the fixed size constraint. The same reasoning works if the nodes connected to w apart from v were not exactly thos in $\mathcal{K}$. In other words, the only possible edges construction that ends up satisfying the fixed size constraint is the one described above. Otherwise the node-to-node inequalities would force to introduce an excess of edges. ∎

Corollary 4.3. *The triangle inequalities are not necessary to formulate the Pricing Problem with fixed size.*

Given $e = ij \in A$, let $c_e = (a_i - a_j)^2$, we can express w_C as:

$$w_C = \sum_{e \in A} \frac{c_e z_e}{\eta}$$

This expression for w_C is the edge weights positive contribution to the objective function in the Pricing Subproblem with fixed cluster size. Now we look for the node weight negative contribution in terms of the variables z_e.

As in [20], let λ_v be the node weight of $v \in N$. Recall that this corresponds to the dual variable solution of the dual of the Master Problem. Then note that if the node v is in the solution C of the Pricing Subproblem, then v will be adjacent to $\eta - 1$ nodes. Otherwise, v will be adjacent to zero nodes. This means that clearly the expression

$$\frac{\sum\limits_{e \in \delta(v)} z_e}{\eta - 1}$$

is 1 when $v \in C$ and 0 otherwise.
On account of that, we can simply compute the node weight contribution to the objective function as:

$$\sum_{v \in N} \lambda_v \frac{\sum\limits_{e \in \delta(v)} z_e}{\eta - 1}.$$

This way we do not need extra variables on nodes beyond the z variables on edges. In summary, the objective function in the Pricing Subproblem with fixed cluster size η is:

$$\sum_{e \in A} \frac{c_e z_e}{\eta} - \sum_{v \in N} \lambda_v \frac{\sum\limits_{e \in \delta(v)} z_e}{\eta - 1}$$

What leads us to an ILP model of the Pricing Subproblem with fixed cluster size $\eta \geq 2$:

$$\begin{array}{ll} \min \sum_{e \in A} \frac{c_e z_e}{\eta} - \sum_{v \in N} \lambda_v \frac{\sum\limits_{e \in \delta(v)} z_e}{\eta - 1} & \\ \sum_{e \in A} z_e = \frac{\eta(\eta - 1)}{2} & \\ \sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta - 2) z_{ij} \geq 0, & i, j \in N,\ i \neq j \\ z_e \in \{0, 1\}, & e \in A \end{array} \tag{14}$$

Note that Corollary 4.3 does not mean that triangle inequalities are implied by inequalities in (14) when the binary constraints are relaxed: indeed, they may be used as cuts to reinforce the continuous relaxation of (14). In [30] the polyhedral properties of problem (14) have been deeply studied.

5. A Heuristic Solution Method

Summing up the results of Section 4 here we define a heuristic solution algorithm based on a Column Generation scheme which is composed of two phases: the *master phase* solves the Master Problem, use the primal relaxed solution to apply two different heuristic procedures to generate feasible microaggregations; the *pricing phase* which exploits the dual solution of the Master Problem to solve the Pricing Problem, either heuristically or exactly. We divide the presentation of the two phases into the following two subsections.

5.1. The Pricing Phase

The Pricing Problem is solved with the following scheme:

INPUT: dual variables λ ;
OUTPUT: a family π of clusters with negative reduced cost.

`PricingProblem`(λ , π)

for $\eta = k$ to $2k - 1$

(a) `SolveHeuristicPricing`(λ,η,π);
(b) if $\pi \neq \emptyset$ then return
(c) else if $Table2(n, \eta) =' CE'$
(d) then `SolveCompleteEnumeration`(λ,η,π)
(e) else `SolveExactPricing`(λ,η,π)
(f) if $\pi \neq \emptyset$ then return

In the above algorithm, the procedure `SolveHeuristicPricing` generates one or more disjoint clusters with negative reduced cost as follows. It starts by sorting the nodes by descending value of the dual variables λ_v and declaring all nodes as unassigned to any new cluster. Then it initializes a new cluster C with the first unassigned node v. Iteratively it adds an unassigned node with the minimum contribution as increase of the reduced cost. When the cluster C reaches cardinality η, if its reduced cost is negative then C is stored and all nodes in C are removed from the set of unassigned nodes. Then the loop is repeated with the next non clustered node v. The procedure is shown below; the notation $[w : C]$ indicates the set of edges with one endpoint equal to w and the other belonging to C. The routine `SolveHeuristicPricing` returns all the clusters that end up in π.

`SolveHeuristicPricing`(λ, η, π)

1. $\pi = \emptyset$; initialize $U = N$ as the set of unassigned nodes ;
2. Sort nodes in U by λ_v in descending order ;
3. while $|U| \geq \eta$
   (a) $v = first(U)$; $U = U \setminus \{v\}$;
   (b) set $C = \{v\}$
   (c) while ($|C| < \eta$)
      i. find $w \in U$ s.t. $\sum_{e \in [w:C]} \frac{c_e}{\eta} - \frac{\lambda_w}{\eta - 1}$ is minimum
      ii. $C = C \cup \{w\}$;
   (d) if C has negative reduced cost then
      i. $\pi = \pi \cup \{C\}$;
      ii. $U = U \setminus C$;
4. return

If the procedure `SolveHeuristicPricing` is not successful in finding a negative reduced cost cluster, then, depending on the value of $Table2(n, \eta)$ (see Table 2), routine `PricingProblem` calls either the `SolveCompleteEnumeration` routine, which solves the Pricing Subproblem for fixed η by simply enumerating all cluster of size η with nested loops, or the `SolveExactPricing` routine, which considers the formulation (14) and solves it with CPLEX Mixed-Integer Solver. The values in Table 2 indicates which solution algorithm is more efficient between `SolveCompleteEnumeration` and `SolveExactPricing`, depending on a large computational experience performed in [30]. Both `SolveCompleteEnumeration` and `SolveExactPricing` return the cluster with minimum reduced cost if negative. The `PricingProblem` routine is stopped at the first value of η returning a nonempty set π.

$\eta \setminus n$	30	40	50	100	200
3	CE	CE	CE	CE	CE
4	CE	CE	CE	CE	CE
5	CE	CE	CE	CE	CE
6	CE	CE	CE	CE	EP
7	EP	EP	EP	EP	EP
8	EP	EP	EP	EP	EP
9	EP	EP	EP	EP	EP

Table 2: Routine selection for the exact solution of the Pricing Subproblem

5.2. The Master Phase

In Section 4 we presented the general Column Generation scheme for Microaggregation. In Subsection 5.1 we specialized the solution of the Pricing Subproblem to make its solution efficient in practice. In this subsection, we specialize the solution of the Master Problem with the aim of using the information provided by its relaxed solution to get new feasible solutions with large improvements with respect to usual MDAV and V-MDAV heuristics.

The ingredients of this procedure, named `MicroaggregationHeuristicMP`, are two heuristics to find feasible integer solutions from fractional ones.

The first one, called `SimpleRoundingHeuristic`, considers the current fractional solution x of the Master Problem and builds a clustering π as follows: (i) finds the cluster C with the maximum value of x_C and adds it to π; (ii) discards all clusters C' with $C \cap C' \neq \emptyset$; (iii) repeats (i) and (ii) until there are at least $2k$ unassigned elements; (iv) if the number of unassigned nodes is between k and $2k-1$, builds a cluster P with all remaining nodes and adds P to π; (v) else assigns each remaining node u to the cluster in π whose centroid is the closest to u (this step is implemented by routine `AddRemainingNodes` and it is straightforward). We summarize `SimpleRoundingHeuristic` in the following pseudocode:

`SimpleRoundingHeuristic`(x,π^*)

1. $U = N$; $\pi = \emptyset$;
2. while ($|U| \geq 2k$)
   (a) $C = \arg\max\{x_C : C \in \overline{\mathcal{C}}\}$;
   (b) $\pi = \pi \cup \{C\}$;
   (c) $U = U \setminus P$;
   (d) Discard all C' such that $C \cap C' \neq \emptyset$;
3. if $k \leq |U| \leq 2k-1$ then $\pi = \pi \cup \{U\}$;
4. else `AddRemainingNodes`(U,π);
5. if SSE(π) $<$ SSE(π^*) then $\pi^* = \pi$.

The second one, called `RoundingMSTHeuristic`, computes the solution z_{ij} with respect to the egdes (i,j) starting from the current fractional solution x of the Master Problem, i.e., $z_{ij} = \sum_{C:i,j \in C} x_C$, and builds a clustering π by starting from a clustering where each cluster contains only one node, then scans edges in descending order of z_{ij} and glues the clusters to which i and j belong to. We summarize the exact steps of `RoundingMSTHeuristic` in the following pseudocode:

`RoundingMSTHeuristic`(x,π^*)

1. $z_{ij} = \sum_{C:i,j \in C} x_C$ for each $i,j \in N, i < j$;
2. sort z_{ij} in descending order;

3. for $max_{cl} = k+1, \ldots, 2k-1$ repeat the following steps
   (a) define $\pi = \{C_l(i) = \{i\} | i \in N\}$;
   (b) for (i,j) in descending order of z_{ij}
   if $|C_l(i)| + |C_l(j)| \leq max_{cl}$ then
   $\pi = \pi \setminus \{C_l(i), C_l(j)\} \cup \{C_l(i) \cup C_l(j)\}$;
   (c) if $|C| \geq k$ for each $C \in \pi$ and $\text{SSE}(\pi) < \text{SSE}(\pi^*)$ then $\pi^* = \pi$.

The above two heuristics are called at each iteration the main loop of the Master Problem solution. The *master phase* can be summarized with the following pseudocode:

`MicroaggregationHeuristicMP`(π^*)

1. Let π^1 be the clustering computed by MDAV heuristic
2. Let π^2 be the clustering computed by V-MDAV heuristic
3. if $SSE(\pi^1) < SSE(\pi^2)$ then $\pi^* = \pi^1$ else $\pi^* = \pi^2$
4. Initialize $\overline{\mathcal{C}} = \{\pi^1, \pi^2\}$ of the Master Problem (MP)
5. Iterate
   (a) Solve (MP) $\rightarrow (x, \lambda)$ primal and dual solutions
   (b) if x is integer and $SSE(x) < SSE(\pi^*)$ then $\pi^* = x$
   (c) else
   `SimpleRoundingHeuristic`(x,π^*)
   `RoundingMSTHeuristic`(x,π^*)
   (d) `PricingProblem`(λ,π)
   (e) $\overline{\mathcal{C}} = \overline{\mathcal{C}} \cup \pi$
6. until $\pi = \emptyset$
7. $SSE_{HMP} = SSE(\pi^*)$.

At the end of the `MicroaggregationHeuristicMP` the clustering π^* is returned as solution. Note that it cannot be worse than the initial solution provided by the heuristics MDAV and V-MDAV.

6. Computational Tests

The procedure MicroaggregationHeuristicMP described in Subsection 5.2 has been implemented in C++ and tested with data sets extracted from the *CASC* project [4] that contains a widely used data set in the field of *Statistical Disclosure Control* literature. In particular, two microdata sets from this project were considered, the "Tarragona" data set and the "Census" data set, which are widely used in the literature. The "Census" data set was obtained on July 27, 2000 using the Data Extraction System of the U. S. Bureau of the Census. It includes 1080 records with 13 attributes each. The "Tarragona" data set comprises figures of 834 companies in the Tarragona area. It was collected during the year 1995 and includes 13 attributes.

We tested the MicroaggregationHeuristicMP performance extracting subsets with 30, 40, 50, 100, and 200 individuals from both the microdata sets "Tarragona" and "Census". The subsets have been extracted with four different criteria: (i) consecutively, (ii) randomly, (iii) ordering individuals according L_2 norm, (iv) ordering individuals according L_2 distance from the overall centroid. For each size and for each extraction criterion we extracted 5 subsets obtaining a total of 100 instances.

For each of those subsets of individuals we also tested our code considering three different minimal cluster sizes $k = 3, 4, 5$, which are amongst the typical k-anonymity parameters used in the state of art for microaggregation [10].

The gaps of the heuristic algorithms are computed according to the following formula:

$$GAP_H(\%) = 100 \cdot \frac{SSE_H - SSE_{MP}}{SSE_H}, \tag{15}$$

where H may be either MDAV, V-MDAV, `RoundingMSTHeuristic` (MSTH), or `SimpleRoundingHeuristic` (SRH). Computational times are in seconds and are referred to the execution of the overall `MicroaggreagationHeuristicMP` routine.

The data of the experiments here presented are available at:
`http://www.iasi.cnr.it/~gentile/ClaudioGentileFiles/papers/MicroAggregation`

The following Tables 3 and 4 contain a row for each size, k, and extraction criterion. The columns are associated with the four heuristics tested MDAV, V-MDAV, `RoundingMSTHeuristic` (MSTH), and `SimpleRoundingHeuristic` (SRH). In each entry there is the average of gaps and computational times over the 5 instances for each size, k, and extraction criterion. In all the rows of those tables we can see dramatic improvements on the gaps obtained in affordable computational times. The best gaps, which are marked in blue, are mostly obtained with the `RoundingMSTHeuristic`. The column "Ex" reports the number of instances that have been solved exactly, that is the `MicroaggreagationHeuristicMP` ends with an integer solution for the Master Problem. We can see that exact solutions are not rare expecially when the size is small (e.g., 30 and 40) and k is 5.

In conclusion the routine `MicroaggregationHeuristicMP` based on the column generation principle and on two rounding heuristics offers a new effective tool to solve the Microaggregation problem in Statistical Disclosure Control.

Acknowledgements

Author Enric Spagnolo-Arrizabalaga was supported by the ERASMUS+ Traineeship Program with a visiting period at CNR-IASI. Author Claudio Gentile was partly supported by the project PRIN 2015B5F27W funded by the Italian Minister for Education, by the ITN 764759 "MINOA: Mixed-Integer Nonlinear Optimization Applications" of the European Union and by CNR Short-Term Mobility program CNR-STM14-PROT26328. Authors Jordi Castro and Claudio Gentile were supported by the MCIU/AEI/FEDER RTI2018-097580-B-I00 project of the Spanish Ministry of Science, Innovation and Universities.

References

[1] J. M. Abowd, J. Domingo-Ferrer, and V. Torra, "Using Mahalanobis distance-based record linkage for disclosure risk assessment," in *Privacy in Statistical Databases 2006* (J. Domingo-Ferrer and L. Franconi, eds.), vol. 4302 of *Lecture Notes in Computer Science*, (Heidelberg), pp. 233–242, Springer, 2006.

[2] D. Baena, J. Castro, and A. Frangioni, "Stabilized Benders methods for large-scale combinatorial optimization, with application to data privacy," *Management Science*, in press, doi:10.1287/mnsc.2019.3341.

[3] D. Baena, J. Castro, and J. A. González, "Fix-and-relax approaches for controlled tabular adjustment," *Computers & Operations Research*, vol. 58, pp. 41–52, 2015.

[4] R. Brand, J. Domingo-Ferrer, and J. M. Mateo-Sanz, "Reference data sets to test and compare SDC methods for protection of numerical microdata." European Project IST-2000-25069 CASC, http://neon.vb.cbs.nl/casc, 2002.

[5] J. Castro, "A shortest paths heuristic for statistical disclosure control in positive tables," *INFORMS Journal on Computing*, vol. 9, no. 4, pp. 520–533, 2007.

[6] J. Castro, "Recent advances in optimization techniques for statistical tabular data protection," *European Journal of Operational Research*, vol. 216, pp. 257–269, 2012.

Table 3: Computational Results: "Census" data set

n	k	Type	MDAV	V-MDAV	MSTH	SRH	Time	Ex
30	3	cons	23.5	14.7	**5.39**	5.89	0.02	
30	3	rand	21.5	18.9	4.26	**3.59**	0.03	
30	3	L_2n	21.0	15.9	**6.65**	7.22	0.02	
30	3	L_2d	32.6	24.6	**4.45**	4.57	0.02	
40	3	cons	22.7	14.2	**4.26**	4.49	0.04	
40	3	rand	14.9	13.0	**3.31**	**3.31**	0.03	
40	3	L_2n	22.4	18.5	**3.87**	3.89	0.04	
40	3	L_2d	35.2	27.9	**2.57**	3.57	0.05	
50	3	cons	17.0	12.4	**4.28**	**4.28**	0.09	
50	3	rand	21.7	18.7	**4.7**	5.05	0.09	
50	3	L_2n	23.6	22.1	**2.80**	2.82	0.12	
50	3	L_2d	32.5	25.0	**2.34**	2.75	0.08	
100	3	cons	22.2	19.6	**3.05**	3.27	2.33	
100	3	rand	21.9	18.1	**3.99**	4.01	2.30	
100	3	L_2n	22.7	19.8	**2.82**	**2.82**	2.34	
100	3	L_2d	32.6	31.0	**3.59**	5.64	2.32	
200	3	cons	18.3	17.9	**7.04**	9.57	74.99	
200	3	rand	22.8	20.6	**2.47**	**2.47**	75.07	
200	3	L_2n	19.8	17.8	**2.27**	**2.27**	75.11	
200	3	L_2d	30.1	25.1	**2.67**	**2.67**	75.25	
30	4	cons	17.3	15.1	**8.50**	9.80	0.13	
30	4	rand	29.8	22.5	**8.55**	9.78	0.16	
30	4	L_2n	29.5	17.2	**3.76**	3.89	0.13	
30	4	L_2d	44.2	26.7	**6.42**	7.60	0.16	
40	4	cons	15.0	13.3	2.94	**2.04**	0.49	1
40	4	rand	15.5	14.2	3.99	**3.66**	0.48	
40	4	L_2n	20.5	18.9	**1.47**	1.48	0.50	2
40	4	L_2d	35.6	22.2	3.71	**3.39**	0.52	
50	4	cons	9.6	10.8	**3.37**	3.41	1.80	
50	4	rand	21.2	17.4	**3.68**	4.33	1.73	
50	4	L_2n	27.1	22.1	**1.64**	3.02	1.86	
50	4	L_2d	38.7	29.7	**3.64**	7.58	1.79	
100	4	cons	17.8	16.0	**3.11**	6.26	75.46	
100	4	rand	17.4	15.5	**3.29**	3.36	75.00	
100	4	L_2n	18.0	17.6	**3.13**	4.93	75.61	
100	4	L_2d	31.2	27.2	**3.22**	7.17	75.22	
200	4	cons	21.2	19.9	**5.19**	13.51	104.88	
200	4	rand	23.0	19.6	**3.86**	5.74	100.96	
200	4	L_2n	18.9	19.3	**3.64**	6.78	93.89	
200	4	L_2d	32.3	28.4	**4.38**	10.69	96.23	
30	5	cons	13.8	9.8	**3.84**	4.41	0.39	2
30	5	rand	23.5	14.2	**5.11**	6.53	0.34	1
30	5	L_2n	15.2	13.7	**1.22**	2.30	0.33	3
30	5	L_2d	32.0	23.2	**3.19**	**3.19**	0.34	1
40	5	cons	13.6	10.3	**1.55**	2.73	1.43	1
40	5	rand	10.3	9.0	**3.62**	3.86	1.25	
40	5	L_2n	14.7	12.5	**0.61**	**0.61**	1.29	3
40	5	L_2d	30.3	26.7	**4.15**	**4.15**	1.40	1
50	5	cons	19.4	18.6	**5.07**	5.98	4.64	1
50	5	rand	19.0	14.8	2.47	**2.38**	4.17	
50	5	L_2n	18.9	18.9	1.94	**1.89**	4.26	1
50	5	L_2d	31.1	23.0	**4.2**	**4.2**	4.37	
100	5	cons	17.8	15.5	**3.21**	3.30	167.14	
100	5	rand	18.7	17.9	**3.26**	3.35	152.96	
100	5	L_2n	20.6	17.7	**2.68**	5.72	155.88	
100	5	L_2d	35.2	25.2	**2.69**	2.98	152.99	
200	5	cons	19.6	17.7	**2.26**	4.90	1823.79	
200	5	rand	22.6	20.2	**4.37**	10.32	2125.44	
200	5	L_2n	21.4	19.9	**1.96**	3.88	1739.64	
200	5	L_2d	38.2	33.1	**3.97**	8.91	1728.77	

[7] J. Castro, A. Frangioni, and C. Gentile, "Perspective reformulations of the CTA problem with L2 distances," *Operations Research*, vol. 62, no. 4, pp. 891–909, 2014.

[8] J. Castro and J. A. González, "A linear optimization based method for data privacy in statistical tabular data," *Optimization Methods and Software*, vol. 34, pp. 37–61, 2019.

[9] T. Dalenius and S. Reiss, "Data-swapping: a technique for disclosure control (extended abstract)," in *Proc. ASA Section on Survey Research Methods*, (Washington DC), pp. 191–194, American Statistical Association, 1978.

[10] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, 2002.

[11] J. Domingo-Ferrer and V. Torra, "A quantitative comparison of disclosure control methods for microdata," in *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies* (P. Doyle, J. Lane, J. Theeuwes, and L. Zayatz, eds.), (Amsterdam), pp. 111–134, North-Holland, 2001.

[12] J. Domingo-Ferrer and V. Torra, "Ordinal, continuous and heterogeneous k-anonimity through microaggregation," *Data Mining and Knowledge Discovery*, vol. 11, pp. 195–212, 2005.

Table 4: Computational Results: "Tarragona" data set

n	k	Type	MDAV	V-MDAV	MSTH	SRH	Time	Ex
30	3	cons	10.8	7.9	**1.00**	**1.00**	0.03	
30	3	rand	3.2	3.0	**0.43**	**0.43**	0.02	1
30	3	L_2n	13.0	11.9	2.15	**1.93**	0.02	
30	3	L_2c	20.8	16.2	**2.82**	**2.82**	0.02	2
40	3	cons	10.3	10.4	**5.88**	**5.88**	0.04	
40	3	rand	3.4	3.3	**0.07**	0.09	0.03	
40	3	L_2n	16.9	12.9	**1.42**	**1.42**	0.05	
40	3	L_2c	25.1	16.3	**2.89**	**2.89**	0.04	
50	3	cons	10.9	9.3	**6.04**	6.11	0.10	
50	3	rand	3.4	3.3	**0.29**	0.33	0.16	
50	3	L_2n	13.4	12.9	**1.54**	1.91	0.10	
50	3	L_2c	30.6	23.9	**1.80**	1.82	0.09	
100	3	cons	7.1	5.9	**2.16**	**2.16**	2.36	
100	3	rand	3.1	3.2	**0.16**	0.39	2.48	
100	3	L_2n	13.9	12.5	**1.39**	**1.39**	2.56	
100	3	L_2c	24.3	19.0	**1.11**	**1.11**	2.36	
200	3	cons	4.1	3.7	**0.59**	1.17	75.31	
200	3	rand	2.8	2.4	**1.10**	2.18	75.47	
200	3	L_2n	13.4	12.6	**3.90**	9.27	75.46	
200	3	L_2c	21.8	23.3	**4.35**	20.31	75.49	
30	4	cons	8.4	6.1	1.00	**0.93**	0.16	
30	4	rand	6.6	5.2	**0.15**	**0.15**	0.13	
30	4	L_2n	20.1	15.1	**2.19**	2.59	0.13	
30	4	L_2c	31.1	21.6	**2.94**	3.08	0.16	
40	4	cons	7.3	5.7	2.83	**2.74**	0.51	
40	4	rand	1.7	1.9	**0.91**	0.92	0.51	
40	4	L_2n	13.1	10.8	**0.11**	**0.11**	0.51	3
40	4	L_2c	17.1	14.6	1.86	**1.85**	0.54	1
50	4	cons	8.6	8.0	**2.28**	2.34	1.83	
50	4	rand	5.1	4.9	**0.38**	**0.38**	1.90	
50	4	L_2n	13.6	13.5	**1.06**	1.11	1.90	
50	4	L_2c	22.3	20.7	**3.95**	4.06	1.78	
100	4	cons	4.2	4.8	**0.45**	0.47	75.83	
100	4	rand	3.1	3.0	**0.90**	2.57	78.34	
100	4	L_2n	13.3	13.0	**2.92**	5.77	75.60	
100	4	L_2c	21.8	16.8	**2.47**	7.01	75.55	
200	4	cons	5.5	5.2	**1.11**	5.10	101.44	
200	4	rand	1.9	1.9	**0.88**	1.85	108.76	
200	4	L_2n	17.7	16.1	**3.98**	13.82	99.94	
200	4	L_2c	24.3	23.5	**3.94**	17.53	105.14	
30	5	cons	4.7	3.5	**0.80**	0.93	0.43	2
30	5	rand	3.7	3.3	**0.46**	**0.46**	0.37	2
30	5	L_2n	12.3	12.4	0.25	**0.24**	0.36	2
30	5	L_2c	25.3	18.9	0.62	**0.48**	0.38	1
40	5	cons	8.0	8.1	0.87	**0.84**	1.46	1
40	5	rand	2.2	1.9	**0.23**	0.64	1.47	1
40	5	L_2n	13.8	11.2	1.64	**1.50**	1.52	
40	5	L_2c	13.9	15.1	**0.78**	**0.78**	1.45	1
50	5	cons	9.7	8.5	**1.04**	**1.04**	4.51	
50	5	rand	1.2	1.2	**0.13**	**0.13**	4.65	2
50	5	L_2n	13.4	13.0	**2.2**	2.33	4.70	1
50	5	L_2c	20.8	17.1	**1.79**	2.06	4.92	
100	5	cons	6.2	5.0	**0.88**	1.71	168.10	
100	5	rand	1.2	1.0	**0.05**	0.06	169.31	
100	5	L_2n	13.5	11.7	**0.81**	1.08	159.60	
100	5	L_2c	18.5	16.4	**3.72**	9.50	165.90	
200	5	cons	6.8	6.8	**1.25**	4.27	2086.33	
200	5	rand	2.1	2.1	**0.58**	1.85	2935.79	
200	5	L_2n	19.9	19.8	**2.67**	10.02	2187.26	
200	5	L_2c	24.1	20.8	**2.63**	14.77	3076.78	

[13] M. Fischetti and J. J. Salazar-González, "Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control," *Mathematical Programming*, vol. 84, no. 2, pp. 283–312, 1999.

[14] M. Fischetti and J. J. Salazar-González, "Solving the cell suppression problem on tabular data with linear constraints," *Management Science*, vol. 47, no. 7, pp. 1008–1027, 2001.

[15] J. Ghosh and A. Liu, "K-Means," in *The Top Ten Algorithms in Data Mining*, (Boca Raton), pp. 21–35, Taylor & Francis, 2009.

[16] J. A. González and J. Castro, "A heuristic block coordinate descent approach for controlled tabular adjustment," *Computers & Operations Research*, vol. 38, pp. 1826–1835, 2011.

[17] S. Hansen and S. Mukherjee, "A polynomial algorithm for optimal univariate microaggregation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, 2003.

[18] M. S. Hernández-García and J. J. Salazar-González, "Enhanced controlled tabular adjustment," *Computers & Operations Research*, vol. 43, pp. 61–67, 2014.

[19] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. S. Nordholt, K. Spicer, and P.-P. de Wolf, *Statistical Disclosure Control*. Chichester: Wiley, 2012.

[20] X. Ji and J. E. Mitchell, "Branch-and-price-and-cut on the clique partitioning problem with minimum clique size requirement," *Discrete Optimization*, vol. 4, no. 1, pp. 87–102, 2007.

[21] M. T. Melo, S. Nickel, and F. S. da Gama, "Facility location and supply chain management–A review," *European Journal of Operational Research*, vol. 196, pp. 401–412, 2009.

[22] R. Moore, "Controlled data-swapping techniques for masking public use microdata," tech. rep., U.S. Bureau of the Census Statistical Research Division, 1996.

[23] K. Muralidhar and R. Sarathy, "Data shuffling: A new masking approach for numerical data," *Management Science*, vol. 52, pp. 658–570, 2006.

[24] A. Oganian and J. Domingo-ferrer, "On the complexity of optimal microaggregation for statistical disclosure control," *Statistical Journal of the United Nations Economic Comission for Europe*, vol. 18, pp. 345–354, 2001.

[25] A. J. Sage and S. E. Wright, "Obtaining cell counts for contingency tables from rounded conditional frequencies," *European Journal of Operational Research*, vol. 250, no. 1, pp. 91–100, 2016.

[26] J. J. Salazar-González, "Mathematical models for applying cell suppression methodology in statistical data protection," *European Journal of Operational Research*, vol. 154, pp. 740–754, 2004.

[27] J. J. Salazar-González, "Controlled rounding and cell perturbation: Statistical disclosure limitation methods for tabular data," *Mathematical Programming*, vol. 105, pp. 583–603, 2006.

[28] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[29] A. Solanas and A. Martínez-Ballesté, "V-MDAV: A multivariate microaggregation with variable group size," in *Proc. COMPSTAT Symp. IASC*, pp. 917–925, 2006.

[30] E. Spagnolo, "On the use of integer programming to pursue optimal microaggregation," Master's thesis, School of Mathematics and Statistics, Universitat Politècnica de Catalunya, 2016.