Thirty years of optimization-based SDC methods for tabular data

Jordi Castro

Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya, Jordi Girona 1–3, 08034, Barcelona, Catalonia.

E-mail: `jordi.castro@upc.edu`

Received 3 March 2022; received in revised form 18 July 2022; accepted 30 July 2022

Abstract. In 1966 Bacharach published in Management Science a work on matrix rounding problems in two-way tables of economic statistics, formulated as a network optimization problem. This is likely the first application of optimization/operations research for statistical disclosure control (SDC) in tabular data. Years later, in 1982, Cox and Ernst used the same approach in a work in INFOR for a similar problem: controlled rounding. And thirty years ago, in 1992, a paper by Kelly, Golden and Assad appeared in Networks about the solution of the cell suppression problem, also using network optimization. Cell suppression was used for years as the main SDC technique for tabular data, and it was an active field of research which resulted in several lines of work and many publications. The above are some of the seminal works on the use of optimization methods for SDC when releasing tabular data. This paper discusses some of the research done this field since then, with a focus on the approaches that were of practical use. It also discusses their pros and cons compared to recent techniques that are not based on optimization methods.

Keywords. Linear optimization, mixed integer linear optimization, network optimization, statistical disclosure control, cell suppression, controlled adjustment, controlled perturbation

1 Introduction

Statistical Disclosure Control (SDC) comprises the set of methods for preserving individual and confidential information when releasing data. It is a main concern of institutions in charge of publishing statistical data (mainly, national statistical agencies). Data are released in the form of either microdata or tabular data. A microdata set is a matrix of p individuals and d variables, where entry (i, j) provides the value of variable j for individual i. Variables can be categorical or numerical. Tabular data is obtained by crossing one or more categorical variables. Cell values can show either the number of individuals (named *respondents* in SDC), or sum of values for another numerical variable; these two groups of tables are named, respectively, *frequency* (or *contingency*), and *magnitude* tables.

Any table can be modeled as a vector of n values (also named *cell values*) $a_i, i = 1, \dots, n$ that satisfy a set of m linear relations $Aa = b$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. In practice cell values are usually between lower and upper bounds $l_i, u_i, i = 1, \dots, n$. If the table is positive then $l_i \geq 0, i = 1, \dots, n$. Each row of matrix $A = (a_{ij}), i = 1, \dots, m, j = 1, \dots, n$ is related to a table linear relation, with coefficients $a_{ij} \in \{1, 0, -1\}$. Each equation contains at most a coefficient -1, which corresponds to the marginal or total cell of the equation. The linear relations $Aa = b$ admit alternative (usually more efficient) representations for some classes of tables:

- The linear equations of two-dimensional (or two-way tables) can be formulated as a bipartite graph, allowing the application of fast network optimization algorithms, such as minimum cost network flows, or shortest paths [1]. This property, first observed in [2], has been applied in several works [6, 10, 20, 31, 34, 43].
- The linear equations of three-dimensional tables correspond to a cube (tensor) of data. They can be modeled as a multicommodity network, where each slice of the cube for one of the dimensions provide a bipartite graph associated to a two-dimensional table. Exploiting this fact, initially noticed in [7], significant computational savings can be obtained [8, 14]. A previous attempt to deal with three-dimensional tables appears in [25].
- Two dimensional tables where one of the two categorical variables has a hierarchical structure (i.e., country→region→city→zip code) is a particular case of *hierarchical table* named *1H2D* (that is, two-dimensional table with one hierarchical variable). These tables also admit a network representation, which was exploited in [10].

A more detailed description on modelling tables can be found in [11].

For a given microdata or tabular data set, let it D, any SDC method can be seen as a map F such that $F(D) = D'$, that is, the original values D are transformed into another dataset D' which is safe and, ideally, with minimum information loss (i.e., D' provides (almost) the same information as D). Moreover, the inverse map $D = F^{-1}(D')$ should be difficult (ideally impossible) to compute in order to guarantee a low disclosure risk.

SDC techniques are classified depending on whether D are microdata or tabular data. Protection techniques for microdata, such as microaggregation [28, 38] or data shuffling [46], are out of the scope of this work. A general description of SDC methods for microdata can be found in the monographs [41, 52]. In general, the number of records in a microdata file p can be much larger than the number of cells n in a table ($p \gg n \gg 0$). On the other hand, SDC techniques for microdata may not need to deal with the linear constraints of tabular data, thus avoiding the solution of linear optimization, mixed integer linear optimization, and even mixed integer quadratic optimization models, which are usually difficult. There are, however, a few exceptions where microdata protection has been combined with constraints. One is the work [51], which presents variants of microaggregation that meet some predefined set of constraints. The second is the recent work [16], where optimal multivariate microaggregation is formulated as a nonlinear combinatorial optimization problem, and a column-generation approach is derived to obtain good feasible solutions with small duality gaps, outperforming current practical heuristics for microaggregation in terms of quality of the solution.

SDC techniques for tabular data can be classified according to different criteria. Depending on whether cell values are modified or not, methods can be classified as *non-perturbative* or *perturbative*. Non-perturbative methods either change the table structure (such as in *recoding*) or hide some cells (*cell suppression* [3, 10, 20, 31, 32, 43]). Perturbative methods modify the cell values; the two representative techniques are *controlled tabular adjustment* (or *controlled perturbation*) [4, 9, 14, 15, 17, 23, 24, 36, 37, 39] and *controlled rounding* [21, 22, 49]. Cell suppression, controlled tabular adjustment, and controlled rounding are based on optimization techniques; the first two focused most of the research on tabular data protection in previous years, and they will be the two techniques mainly discussed in Sections 2.1 and 2.2

An alternative classification criterion is whether the table is first computed and afterwards protected (*post-tabular* methods), or the microdata file is first modified and then the table is built (*pre-tabular* method). Most methods relying on optimization techniques are post-tabular, with just a few exceptions such as the pre-tabular method of [40]. Similarly, some post-tabular approaches do not use optimization methods. Among them, the *cell-key* method [44] is recently being adopted by several national statistical institutes [30]. A discussion comparing cell-key vs optimization-based methods is presented in Section 3.

2 Optimization-based methods for tabular data

SDC techniques for tabular data relying on optimization methods share a common set of parameters, which are needed for the definition of the optimization model. Those parameters are:

- A general table, consisting of a set of n cells and a set of m linear relations $Aa = b$, where $A \in \mathbb{R}^{m \times n}$ is the matrix defining the table structure, $a = (a_1, \ldots, a_n)^T \in \mathbb{R}^n$ is the vector of cell values, and the right-hand side $b \in \mathbb{R}^m$ is usually 0 if the table is additive.
- Upper and lower bounds $u \in \mathbb{R}^n$ and $l \in \mathbb{R}^n$ for the cell values, which are assumed to be known by any attacker: $l \leq a \leq u$ (e.g., $l = 0$, $u = +\infty$ for a positive table).
- Vector of nonnegative weights $w \in \mathbb{R}^n$, associated to either the cell suppressions, or the cell perturbations. That is, $w_i, i = 1, \ldots, n$ is a measure of the cost (or data utility loss) associated to hiding the true value of cell i. If $w_i = 1$ for all $i = 1, \ldots, n$, the same cost is given to any cell; if $w_i = 1/a_i$ a relative cost is considered depending on the cell values; other options are possible, such as, for instance, $w_i = 1/\sqrt{a_i}$. Several cost options were analyzed in [17].
- Set $\mathcal{S} \subseteq \{1, \ldots, n\}$ of sensitive cells, decided in advance by applying some sensitivity rules. More details about sensitivity rules, which are out of the scope of this paper, can be found in [29, 41, 48]
- Lower and upper protection levels for each sensitive cell lpl_s and upl_s $s \in \mathcal{S}$ (usually either a fraction of a_s or directly obtained from the sensitivity rules). Sliding protection levels can also be considered, as in [32].

Next two subsections define the cell suppression and controlled tabular adjustment optimization models, discussing some of the solution approaches developed during the last thirty years.

2.1 Cell suppression problem

The *cell suppression problem* (CSP) aims at finding a set $\mathcal{C} \in \{1, \ldots, n\}$ of cells (named *complementary* cells) to be removed—in addition to the set of sensitive cells $\mathcal{S}$—such that for all $s \in \mathcal{S}$

$$\underline{a_s} \leq a_s - lpl_s \quad \text{and} \quad \overline{a_s} \geq a_s + upl_s, \tag{1}$$

where $\underline{a_s}$ and $\overline{a_s}$ are computed (once the pattern of suppressed cells $\mathcal{S}\cup\mathcal{C}$ has been obtained) by the solution of the two following linear optimization problems:

$$\begin{array}{rll} \underline{a_s} = & \min\limits_{x} & x_s \\ & \text{s. to} & Ax = b \\ & & l_i \leq x_i \leq u_i \quad i \in \mathcal{S}\cup\mathcal{C} \\ & & x_i = a_i \quad i \notin \mathcal{S}\cup\mathcal{C} \end{array} \quad \text{and} \quad \begin{array}{rll} \overline{a_s} = & \max\limits_{x} & x_s \\ & \text{s. to} & Ax = b \\ & & l_i \leq x_i \leq u_i \quad i \in \mathcal{S}\cup\mathcal{C} \\ & & x_i = a_i \quad i \notin \mathcal{S}\cup\mathcal{C}. \end{array} \tag{2}$$

Values $\underline{a_s}$ and $\overline{a_s}$ are considered the minimum and maximum values that an attacker can estimate for sensitive cell $s \in \mathcal{S}$ once the (suppressed) table is released. The recent work [45] shows that, in practice, the interval protection is narrower that the one given by $[\underline{a_s}, \overline{a_s}]$ if the attacker knows the algorithm that has been used to protect the table.

The classical model for CSP, originally formulated in [43], considers two sets of variables: (i) $y_i \in \{0,1\}, i = 1,\dots,n$, is 1 if cell i has to be suppressed, and 0 otherwise; (ii)) two auxiliary vectors $x^{l,s} \in \mathbb{R}^n$ and $x^{u,s} \in \mathbb{R}^n$, for all $s \in \mathcal{S}$. The resulting model is:

$$\begin{array}{ll} \min\limits_{y,x^{l,s},x^{u,s}} & \sum\limits_{i=1}^{n} w_i y_i \\ \text{s. to} & \left.\begin{array}{rlll} & Ax^{l,s} & = 0 & \\ (l_i - a_i)y_i \leq & x_i^{l,s} & \leq (u_i - a_i)y_i & i = 1,\dots,n \\ & x_s^{l,s} & \leq -lpl_s & \\ & & & \\ & Ax^{u,s} & = 0 & \\ (l_i - a_i)y_i \leq & x_i^{u,s} & \leq (u_i - a_i)y_i & i = 1,\dots,n \\ & x_s^{u,s} & \geq upl_s & \end{array}\right\} \forall\, s \in \mathcal{S} \\ & y_i \in \{0,1\} \quad i = 1,\dots,n. \end{array} \tag{3}$$

When $y_i = 1$, the inequality constraints of (3) with both right- and left-hand sides impose bounds on the deviations $x_i^{l,p}$ and $x_i^{u,p}$ for cell i, such that the solution to problems (2) would satisfy (1) When $y_i = 0$ (that is, cell i is not suppressed), the inequality constraints of (3) impose that $x_i^{l,s} = 0$ and $x_i^{u,s} = 0$, that is, the cell is published and any attacker knows its true value. Formulation (3) gives rise to a mixed integer linear optimization problem (MILP) of n binary variables, $2n|\mathcal{S}|$ continuous variables, and $2(m+2n+1)|\mathcal{S}|$ constraints. For instance, for a table of 4000 cells, 1000 sensitive cells, and 2500 linear relations, the formulation has 8000000 continuous variables, 4000 binary variables, and 21000000 constraints.

Solution approaches developed for (3) can be broadly classified in three groups: (i) heuristic methods based on network optimization; (ii) exact approaches based on optimization decomposition methods (mainly Benders decomposition); and (iii) other heuristics (some of them using the previous optimization-based approaches for subproblem solution):

- Heuristic approaches based on network optimization [1] find a (hopefully good) feasible solution to (3) for tables that accept a network representation, that is, two-dimensional and 1H2D tables. The seminal paper [43] formulated and found a feasible solution for (3) in two-dimensional tables by solving a sequence of minimum cost network flow problems. The work [25] attempted to extend this idea to three-dimensional tables, but as mentioned above, three-dimensional tables are formulated as multicommodity flows, so they cannot be solved with a generalization of [43]. An alternative efficient procedure based on shortest paths was suggested in [6], but it was only valid for general tables (that is, cell values can take any value, either positive, zero or negative). Other variants based on minimum cost network flow problems were introduced in [20] and [7]. Some of the above ideas were sensibly combined in the approach of [10], which is based on shortest paths and it is valid for tables with positive cell values. Approaches based on shortest paths are in general much more efficient than those based on minimum cost network flows. However, the main drawback of all the heuristics based on network optimization is that they are only valid for tables accepting a network representation. Some of the above network optimization heuristics were outlined in [11].

- The optimal solution of problem (3) using a general state-of-the-art MILP solver can be impractical due to an excessive computation time. However, due to its structure, (3) can be tackled by a Benders decomposition [5], a standard decomposition technique in optimization. Benders decomposition is a cut generation procedure that, in its classical form, alternates between the solution of a master problem in the binary variables $y_i \in \{0,1\}, i = 1,\dots,n$—which provides a suppression pattern—, and $|\mathcal{S}|$ subproblems (one per sensitive cell) that check whether the sensitive cells are protected. If they are, then the procedure stops with an optimal solution; otherwise, the protection of some sensitive cell is violated, obtaining a *feasibility cut* which is added to the master problem for the next iteration.

  Benders decomposition was first applied for two-dimensional tables in [31], and later extended to general tables in [32]. In these two works, the classical version of Benders decomposition was not used, as the feasibility cuts were added within a generic branch-and-cut algorithm. This allowed the optimal solution of nontrivial CSP problems for the first time. Recently, a stabilized version of Benders decomposition was introduced in [3]. This stabilized version was competitive with that of [32]. In particular, as reported in [3], in a set of synthetic 1H2D tables the stabilized Benders decomposition always obtained a solution with a small duality (i.e., optimality) gap, whereas the approach of [32] either obtained solutions of similar gap requiring more CPU time, or exhausted the time limit without a feasible solution.

- We mention two other practical heuristics developed by national statistical institutes to efficiently deal with large tables: *hypercube* [35] and *Hitas*. [26]. The hypercube heuristic computes simple protection patterns: in a k-dimensional table, it finds a suppression pattern of $2^k - 1$ complementary cells for each sensitive cell. Linked tables are decomposed in subtables, and the hypercube is repeatedly applied; this variant is named the *GHMITER* heuristic. Hitas also decomposes a k-dimensional hierarchical table ($k \leq 3$) in a tree of $(k-l)$-dimensional non-hierarchical subtables, $l = 0,\dots,k-1$, and locally protects them by the approach of [31]. These two heuristics are in general much faster than optimal methods based on Benders decomposition, and can deal with general linked tables (unlike the network optimization based heuristics). However, since some linking constraints between subtables are removed, the final solution can not be feasible. In addition, hypercube tends to over-suppress cells. It is worth noting that, in spite of their infeasibility issues, both hypercube and Hitas provide solutions that avoid the *singleton* issue [42, 47]: this happens, for instance, when two cells with a single respondent (a singleton) appear in the same table relation, such that any of them can compute the other's contribution.

Some of the above approaches for CSP are implemented in the τ-Argus package for SDC

Table 1: Results for 1H2D tables from τ-Argus distribution, obtained crossing variables "IndustryCode"×"Size", and reporting information for either "Var1" or "Var2" (table from [11]).

			Benders		shortest paths		Hitas		hypercube	
Instance	n	$\lvert\mathcal{S}\rvert$	f^*	CPU	f^*	CPU	f^*	CPU	f^*	CPU
Var1	6399	657	18.7e+6	$604^{(1)}$	20.4e+6	1	25.7e+6	4	54.2e+6	4
Var2	6399	1018	6.85e+6	$605^{(1)}$	8.26e+6	2	8.75e+6	6	14.3e+6	3

$^{(1)}$ stopped after default τ-Argus time limit of 10 minutes reached

in tabular data [27], namely, the optimal method based on Benders cuts of [32], the shortest-path heuristic of [10], and the Hitas and hypercube heuristics of, respectively, [26] and [35]. Table 1 (from [11]) shows the computational results obtained with the four methods, for two tables generated using the data files accompanying the τ-Argus distribution. For each method the table provides: columns f^* with the objective function of the optimization problem solved, that is, the value of suppressed cells (thus, the lower, the better); and the solution time (columns "CPU"). The number of cells n and number of sensitive cells $|\mathcal{S}|$ are also given. It is observed that the optimal Benders decomposition provides better solutions but exhausted the 10 minutes time limit. On the other hand, the shortest-path heuristic got decent solutions in 1–2 seconds. The hypercube and Hitas are also very fast, but, mainly for hypercube, f^* was much larger.

2.2 Controlled tabular perturbation/adjustment

Unlike CSP, *controlled tabular adjustment* (also known as *minimum-distance controlled tabular adjustment* or simply *CTA*), is a perturbative method, the purpose of which is to find the closest safe table to the original table. CTA was introduced in the manuscript [24] and, independently in [9] (in the latter it was named minimum-distance controlled perturbation method). In CTA closeness is measured as the norm (usually, ℓ_1 or ℓ_2 (Euclidean) norm) of the perturbations added to the original cell values. As it was empirically shown in [12], CTA has a low disclosure risk, that is, estimates of sensitive cells obtained by an attacker are far enough from the original cell values.

Given a table $a \in \mathbb{R}^n, Aa = b$, CTA computes an alternative safe and feasible table $x \in \mathbb{R}^n, Ax = b$, which is closest to a according to some particular distance $\ell_{(w)}$ based on cell weights $w \in \mathbb{R}^n$. In CTA *safe* means that the new values of sensitive cells x_s are outside the protection interval $[a_s - lpl_s, a_s + upl_s]$ for all $s \in \mathcal{S}$. The optimization problem to be solved is:

$$\begin{array}{ll} \min\limits_{x} & ||x-a||_{\ell(w)} \\ \text{s. to} & Ax = b \\ & l \le x \le u \\ & x_s \le a_s - lpl_s \text{ or } x_s \ge a_s + upl_s \quad s \in \mathcal{S}. \end{array} \tag{4}$$

Defining cell deviations $z = x - a$, $l_z = l - a$ and $u_z = u - a$, (4) can be reformulated as:

$$\begin{array}{ll} \min\limits_{z} & ||z||_{\ell(w)} \\ \text{s. to} & Az = 0 \\ & l_z \le z \le u_z \\ & z_s \le -lpl_s \text{ or } z_s \ge upl_s \quad s \in \mathcal{S}. \end{array} \tag{5}$$

The "or" constraints of (5) can be modeled using binary variables $y_s \in \{0,1\}$, $s \in \mathcal{S}$, such that $y_s = 1$ if cell s is "upper protected" (i.e, $z_s \ge upl_s$), and $y_s = 0$ if it is "lower protected"

$(z_s \leq -lpl_s)$. For distance ℓ_1, the resulting MILP formulation is

$$\begin{array}{ll} \min\limits_{z^+,z^-,y} & \sum\limits_{i=1}^{n} w_i(z_i^+ + z_i^-) \\ \text{s. to} & A(z^+ - z^-) = 0 \\ & 0 \leq z_i^+ \leq u_{z_i} \quad i \notin \mathcal{S} \\ & 0 \leq z_i^- \leq -l_{z_i} \quad i \notin \mathcal{S} \\ & upl_i y_i \leq z_i^+ \leq u_{z_i} y_i \quad i \in \mathcal{S} \\ & lpl_i(1-y_i) \leq z_i^- \leq -l_{z_i}(1-y_i) \quad i \in \mathcal{S} \\ & y_i \in \{0,1\} \quad i \in \mathcal{S}. \end{array} \tag{6}$$

where z_i, $i = 1, \dots, n$, is split as $z_i = z_i^+ - z_i^-$, such that $|z_i| = z_i^+ + z_i^-$. Problem (6) has $|\mathcal{S}|$ binary variables, $2n$ continuous variables and $m + 4|\mathcal{S}|$ constraints.

For distance ℓ_2 CTA results in a MIQP (mixed integer quadratic optimization problem), whose formulation is

$$\begin{array}{ll} \min\limits_{z,y} & \sum\limits_{i=1}^{n} w_i z_i^2 \\ \text{s. to} & Az = 0 \\ & l_{z_i} \leq z_i \leq u_{z_i} \quad i \notin \mathcal{S} \\ & l_{z_i}(1-y_i) + upl_i y_i \leq z_i \leq u_{z_i} y_i - lpl_i(1-y_i) \quad i \in \mathcal{S} \\ & y_i \in \{0,1\} \quad i \in \mathcal{S}. \end{array} \tag{7}$$

Problem (7) has n continuous variables, $|\mathcal{S}|$ binary variables, and $m + 2|\mathcal{S}|$ constraints. Although it is smaller in size than (6), in general is a much more difficult problem because of the quadratic objective function. Alternative, stronger, perspective reformulations for (7) were presented in [15].

CTA optimization problems (either linear (6) or quadratic (7)) are much smaller than the CSP formulation (3). This allows the solution of CTA using state-of-the-art solvers. An implementation of (6) using the Xpress and CPLEX solvers [19] was added to the τ-Argus package [27]. Results reported in [11] show that (real-world) instances of similar size as those in Table 1 were optimally solved with this CTA implementation in less than one minute. Much larger real-world tables, however, required a few hours for optimal solutions.

To avoid such costly executions with CTA for large tables, alternative solution approaches have been developed. For instance, a Benders decomposition for CTA was introduced in [13]; it was shown to be effective for two-dimensional tables, while for more complex tables still required large CPU times. Heuristic block coordinate descent and fix-and-relax heuristics were successfully attempted, respectively, in [37] and [4]. For most tables, they provided a good solution in a fraction of the time required by state-of-the-art solvers based on branch-and-cut algorithms. Another heuristic method was suggested in [36], but it was only tested in very small tables. The difficult CTA problem with Euclidean distances in the objective function was considered in [15]; the perspective reformulations used were more efficient than state-of-the-art MIQP solvers. Alternative CTA-like procedures were presented in [39] and [18]. In particular, the approach of [18] considers a linear (instead of MILP) formulation of CTA, where binary variables (i.e., the direction of protection for sensitive cells) are a priori fixed, and possible infeasibility issues are dealt with by modifying bounds of variables and right-hand-sides of constraints by means of a multiobjective formulation. This continuous CTA approach is much more efficient than the MILP-CTA formulation.

3 Optimization vs noise-addition based approaches

Recently, some SDC approaches for tabular data not relying on optimization methods have been developed. The most relevant of these techniques is the *cell-key* method, suggested in [33, 44, 50]. Although both cell-key and optimization-based approaches are post-tabular methods, there is a significant difference between them: cell-key allows computing the changes to cell values while the table is being built, while optimization-based techniques require the complete table to start the optimization procedure. In this sense, cell-key can be considered a "partial" post-tabular method, whereas optimization-based approaches are "complete" post-tabular techniques.

Cell-key involves that a random number, named the key, is attached to each record of the microdata file. For consistency, this key is unique for all the tables derived from this microfile. When a (for instance, frequency) table is built crossing categorical variables of the microdata file, for each cell of the table we compute the sum of keys (modulo some number) of the records comprised in the cell. This value is the cell-key. The released cell value will be the original cell value plus a perturbation. This perturbation is obtained from a double-entry precomputed table of perturbations, which for certain classes of original cell values and cell-key provides the particular perturbation to be used. This method is consistent, in the sense that if the same cell appears in more than one table, the perturbation—and thus the released cell value—will always be the same. Cell-key has been recently added to τ-Argus package.

Compared to optimization-based approaches, such as CSP or CTA, cell-key is obviously much faster because it completely ignores the table linear relations $Ax = b$ (it just can exactly satisfy the cell bounds, e.g., $x \geq 0$ for frequency tables). Satisfying linear relations $Ax = b$ is indeed what makes difficult the application of optimization-based approaches such as CTA or CSP; if the linear relations are removed, then it is possible to imagine new (much simpler and faster) approaches. Perturbations added by the cell-key method guarantee consistency if the same cell appears in different tables, but may not preserve the additivity of the table, thus $Ax \neq b$. In addition, ignoring table relations increases the disclosure risk. For those reasons, some approaches based on cell-key, such as [30], apply a linear optimization (CTA-like) postprocess to recover table additivity.

In addition to table additivity, there is a second related important feature which is not tackled by cell-key: inconsistencies can appear if users compute the same aggregated values by means of linear combinations of different sets of noisy cell values. Those inconsistencies are even obtained if linear additivity is restored using the approach of [30] in linked tables. To avoid those inconsistencies noise-addition techniques such as cell-key should rely on some algorithm for linear systems of equations $Ax = b$. But in this case, cell-key would probably not be too different from CSP or CTA in terms of efficiency.

4 Conclusions

We have reviewed some of the most significant optimization-based approaches for SDC in tabular data of the last 30 years, with a focus on CSP and CTA. Those methods, which have been extensively used in past years, have the downside of large computational times for complex and large tables (in many cases, impractical times if an optimal solution is required). To overcome such a drawback, alternative methods not based on optimization, such as cell-key, have recently been proposed, and are being extensively used. These approaches are very fast, basically because they remove the linear relations inherent to tables. If data releasers consider that table relations $Ax = b$ can be neglected, this, of course, opens the door to new and faster techniques, likely not based on optimization: one of the strengths of optimization-based techniques is their ability to deal with linear relations.

Acknowledgements

This work has been supported by MCIN/AEI/FEDER grant RTI2018-097580-B-I00.

References

[1] Ahuja, R.K., Magnanti, T.L., Orlin, J.B. (1993) Network flows. Theory, Algorithms and Applications, Prentice Hall, Upper Saddle River, NJ

[2] Bacharach, M. (1966) Matrix rounding problems, Management Science 9 732–742

[3] Baena, D., Castro, J., Frangioni, A. (2020) Stabilized Benders methods for large-scale combinatorial optimization, with application to data privacy, Management Science 66 3051–3068

[4] Baena, D., Castro, J., González, J.A. (2015) Fix-and-relax approaches for controlled tabular adjustment, Computers and Operations Research 58 41-52

[5] Benders, J.F. (2005), Partitioning procedures for solving mixed-variables programming problems, Computational Management Science 2 3–19. English translation of the original paper appeared in (1962) Numerische Mathematik 4 238–252

[6] Carvalho, F.D., Dellaert, N.P., Osório, M.D. (1994) Statistical disclosure in two-dimensional tables: general tables, Journal of the American Statistical Association 89 1547–1557

[7] Castro, J. (2002) Network flows heuristics for complementary cell suppression: an empirical evaluation and extensions, Lecture Notes in Computer Science 2316 59–73

[8] Castro, J. (2005) Quadratic interior-point methods in statistical disclosure control, Computational Management Science 2 107-121

[9] Castro, J. (2006) Minimum-distance controlled perturbation methods for large-scale tabular data protection, European Journal of Operational Research 171 39–52

[10] Castro, J. (2007) A shortest-paths heuristic for statistical data protection in positive tables, INFORMS Journal on Computing 19 520-533

[11] Castro, J. (2012) Recent advances in optimization techniques for statistical tabular data protection, European Journal of Operational Research 216 257–269

[12] Castro, J. (2012) On assessing the disclosure risk of controlled adjustment methods for statistical tabular data, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20 921-941

[13] Castro, J., Baena, D. (2008) Using a mathematical programming modeling language for optimal CTA, Lecture Notes in Computer Science 5262 1–12

[14] Castro, J., Cuesta, J. (2013) Solving L_1-CTA in 3D tables by an interior-point method for primal block-angular problems, TOP 21 25–47

[15] Castro, J., Frangioni, A., Gentile, C. (2014) Perspective reformulations of the CTA problem with L_2 distances, Operations Research 62 891–909

[16] Castro, J., Gentile, C., Spagnolo-Arrizabalaga, E. (2022) An algorithm for the microaggregation problem using column generation, Computers and Operations Research (to appear)

[17] Castro, J., Giessing, S. (2006) Testing variants of minimum distance controlled tabular adjustment, in Monographs of Official Statistics. Work session on Statistical Data Confidentiality, Eurostat-Office for Official Publications of the European Communities, Luxembourg, 333–343

[18] Castro, J., González, J. A. (2019) A linear optimization based method for data privacy in statistical tabular data, Optimization Methods and Software 34 37–61

[19] Castro, J., González, J.A., Baena, D. (2009) User's and programmer's manual of the RCTA package, Technical Report DR 2009/01, Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya

[20] Cox, L.H. (1995) Network models for complementary cell suppression, Journal of the American Statistical Association 90 1453–1462

[21] Cox, L.H, Ernst, L.R. (1982) Controlled rounding, INFOR 20 423–432

[22] Cox, L.H., George, J.A. (1989) Controlled rounding for tables with subtotals, Annals of Operations Research, 20 141–157

[23] Cox, L.H., Kelly, J.P., Patil, R. (2005) Computational aspects of controlled tabular adjustment: algorithm and analysis, B. Golden, S. Raghavan, E. Wassil, eds. The Next wave in Computer, Optimization and Decision Technologies, Kluwer, Boston, MA, 45–59

[24] Dandekar, R.A., Cox, L.H. (2002) Synthetic tabular data: An alternative to complementary cell suppression, manuscript, Energy Information Administration, US Department of. Energy.

[25] Dellaert, N.P., Luijten, W.A. (1999) Statistical disclosure in general three-dimensional tables, Statistica Neerlandica 53 197–221

[26] de Wolf, P.-P. (2002) HiTaS: A heuristic approach to cell suppression in hierarchical tables, Lecture Notes in Computer Science 2316 74–82

[27] de Wolf, P.-P., Hundepool, A., Giessing, S., Salazar-González, J.J., Castro, J. (2014) τ-Argus User's Manual v4.1, Statistics Netherlands. Available on-line at https://research.cbs.nl/casc/Software/TauManualV4.1.pdf.

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

[29] Domingo-Ferrer, J., Torra, V. (2002) A critique of the sensitivity rules usually employed for statistical table protection, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 10 545–556

[30] Enderle, T., Giessing, S., Tent, R. (2018) Designing confidentiality on the fly methodology—Three aspects, Lecture Notes in Computer Science 11126 28–42

[31] Fischetti, M., Salazar-González, J.J. (1999) Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control, Mathematical Programming 84 283–312

[32] Fischetti, M., Salazar-González, J.J. (2001) Solving the cell suppression problem on tabular data with linear constraints, Management Science 47 1008–1026

[33] Fraser, B., Wooton, J. (2005) A proposed method for confidentialising tabular output to protect against differencing, Joint UNECE/EurostatWork Session on Statistical Data Confidentiality, Geneva, Switzerland

[34] Geurts, J. (1992) Heuristics for cell suppression in tables, Technical report, Statistics Netherlands

[35] Giessing, S., Repsilber, D. (2002) Tools and strategies to protect multiple tables with the GHQUAR cell suppression engine, Lecture Notes in Computer Science 2316 181–192

[36] Glover, F., Cox, L.H., Patil, R., Kelly, J.P. (2011) Integrated exact, hybrid and metaheuristic learning methods for confidentiality protection, Annals of Operations Research 183 47–73

[37] González, J.A., J. Castro, J. (2011) A heuristic block coordinate descent approach for controlled tabular adjustment, Computers and Operations Research 38 1826–1835

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

[39] Hernández, M.S., Salazar-González, J.J. (2014) Enhanced controlled tabular adjustment, Computers and Operations Research 43 61–67

[40] Höhne, J. (2011) SAFE – a method for anonymising the German Census, working paper 16 of the Joint UNECE/Eurostat work session on statistical data confidentiality, Tarragona, 1–10

[41] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Schulte Nordholt, E., Spicer, K., de Wolf, P.-P. (2012) Statistical Disclosure Control, Wiley, Chichester, UK

[42] Jewett, R. (1993) Disclosure analysis for the 1992 economic census, manuscript, Economic Programming Division, U.S. Bureau of the Census, Washington, DC.

[43] Kelly, J.P., Golden, B.L., Assad, A.A. (1992) Cell suppression: disclosure protection for sensitive tabular data, Networks 22 28–55

[44] Marley, J. K., Leaver, V. L. (2011) A method for confidentialising user-defined tables: statistical proper-ties and a risk-utility analysis, Proceedings of 58th World Statistical Congress, 1072–1081

[45] Minami, K., Abe, Y. (2019) Algorithmic matching attacks on optimally suppressed tabular data, Algorithms 12 165

[46] Muralidhar, K., Sarathy, R. (2006) Data shuffling: a new masking approach for numerical data, Management Science 52 658–570

[47] Robertson, D. (2000) Improving Statistics Canada's cell suppression software (CONFID), Proceedings in Computational Statistics (eds. J.G. Bethlehem and P.G.M. Van der Heijden) Physica-Verlag New York 403–408

[48] Robertson, D.A., Ethier, R. (2002) Cell suppression: experience and theory, Lecture Notes in Computer Science 2316 8–20

[49] Salazar-González, J.J. (2006) Controlled rounding and cell perturbation: statistical disclosure limitation methods for tabular data, Mathematical Programming 105 583–603

[50] Thompson, G., Broadfoot, S., Elazar, D. (2013) Methodology for the automatic confidentialisation of statistical outputs from remote servers at the Australian Bureau of Statistics, Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, Ottawa, Canada

[51] Torra, V. (2008) Constrained microaggregation:adding constraints for data editing, *Transactions on Data Privacy* 1 86–104

[52] Willenborg, L., de Waal, T. (eds.) (2000) Lecture Notes in Statistics. Elements of Statistical Disclosure Control 155, Springer, New York