Symmetric vs Asymmetric Protection Levels in SDC Methods for Tabular Data

Daniel Baena, Jordi Castro(✉), and José A. González

Department of Statistics and Operations Research,
Universitat Politècnica de Catalunya, Jordi Girona 1–3,
08034 Barcelona, Catalonia, Spain
danibaena@gmail.com, {jordi.castro,jose.a.gonzalez}@upc.edu

Abstract. Protection levels on sensitive cells—which are key parameters of any statistical disclosure control method for tabular data—are related to the difficulty of any attacker to recompute a good estimation of the true cell values. Those protection levels are two numbers (one for the *lower protection*, the other for the *upper protection*) imposing a *safety interval* around the cell value, that is, no attacker should be able to recompute an estimate within such safety interval. In the symmetric case the lower and upper protection levels are equal; otherwise they are referred as asymmetric protection levels. In this work we empirically study the effect of symmetry in protection levels for three protection methods: cell suppression problem (CSP), controlled tabular adjustment (CTA), and interval protection (IP). Since CSP and CTA are mixed integer linear optimization problems, it is seen that the symmetry (or not) of protection levels affect to the CPU time needed to compute a solution. For IP, a linear optimization problem, it is observed that the symmetry heavily affects to the quality of the solution provided rather than to the solution time.

Keywords: Statistical disclosure control · Tabular data
Cell suppression · Controlled tabular adjustment
Interval protection · Mixed integer linear optimization
Linear optimization

1 Introduction

The three statistical disclosure control methods for tabular data considered in this work (namely: cell suppression problem (CSP) [5,10], controlled tabular adjustment (CTA) [2,4,13], and interval protection (IP) [8,11]) belong to the family of *post-tabular* data protection methods, which modify or suppress table cells once the table have been built (in contrast to pre-tabular methods, which change microdata files, and therefore, although being faster, may not guarantee

Supported by grant MTM2015-65362-R of the Spanish Ministry of Economy and Competitiveness.

J. Domingo-Ferrer and F. Montes (Eds.): PSD 2018, LNCS 11126, pp. 3–12, 2018.
https://doi.org/10.1007/978-3-319-99771-1_1

table additivity if the true values of marginal or total cells want to be preserved). More details can be found in the monograph [14] and the survey [6].

Each method protects sensitive cells in a different way. CSP removes sensitive cells; other additional cells have also to be removed to avoid recomputing the original value of sensitive cells. CSP results in a large and difficult mixed integer linear problem, which can be solved optimally (using Benders decomposition as done in [10]) or heuristically (e.g., using shortest paths for some hierarchical tables as in [5]). IP (or partial cell suppression, which was its original name coined in [11]) can be seen as a linear version of CSP, where cell values are replaced by intervals containing the true value. IP, unlike CSP, is a linear optimization problem, and therefore—at least theoretically—it can be solved in polynomial time by efficient interior-point methods [17]. CTA replaces sensitive values by safe values (i.e., outside the safety interval), thus forcing changes in other cells to preserve the table additivity. CTA is also formulated as a mixed integer linear optimization problem, which can be solved optimally by general purpose solvers [9], or heuristically [2,13]. This work provides a formulation of CSP, CTA and IP from the same set of parameters.

One of the key parameters for the optimization models for CSP, CTA and IP are the *lower and upper protection levels*: these two numbers define a protection interval around the cell value, such that no attacker should be able to obtain an estimation of the true value within such interval. When the lower and upper protection levels are equal, we have a *symmetric* interval around the true value; otherwise we refer to the *asymmetric* case. A priori, asymmetric intervals could benefit the solution of mixed integer linear optimization problems, such as CTA and CSP. Indeed, some results along these lines were obtained in [9] for CTA with quadratic objectives. Another objective of this work is to check if such behaviour is observed for CTA and CSP in the solution of a set of hierarchical tables.

For IP, being a linear optimization model, such symmetry is not expected to provide faster executions. However, as it will be shown in the computational results, the use of asymmetric protection levels is instrumental to avoid the disclosure of the true cell values.

This short paper is organized as follows. Section 2 shows a formulation of CSP, CTA and IP using a unified set of parameters. Section 3 reports and compares the results obtained on a set of generated hierarchical instances, using symmetric and asymmetric protection levels, for the three tabular data protection methods.

2 Formulation of CSP, CTA and IP for Tabular Data

The parameters that define any CSP, CTA or IP instance 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, $a = (a_1, \ldots, a_n)^\top \in \mathbb{R}^n$ 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 for CSP, the cell perturbations for CTA, or the width of interval replacing cells for IP. That is, $w_i, i = 1, \ldots, n$ measures 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, $w_i = 1/\sqrt{a_i}$.
- Set $\mathcal{S} \subseteq \{1, \ldots, n\}$ of sensitive cells, decided in advance by applying some sensitivity rules.
- 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). No sliding protection is considered, unlike in [10].

2.1 Formulation of Cell Suppression Problem (CSP)

CSP aims at finding a set $\mathcal{C}$ of complementary cells to be removed such that for all $s \in \mathcal{S}$

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

$\underline{a_s}$ and $\overline{a_s}$ being defined as

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

The classical model for CSP, originally formulated in [15], considers two sets of variables: (1) $y_i \in \{0, 1\}, i = 1, \ldots, n$, is 1 if cell i has to be suppressed, and 0 otherwise; (2) two auxiliary vectors $x^{l,s} \in \mathbb{R}^n$ and $x^{u,s} \in \mathbb{R}^n$, for all $s \in \mathcal{S}$, to impose as constraints that the solutions to problems (2) would satisfy (1). The resulting model is

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

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; these deviations are prevented when $y_i = 0$, that is, when the cell is published (non-suppressed). Formulation (3) gives rise to a mixed integer linear optimization problem of n binary variables, $2n|\mathcal{S}|$ continuous variables, and $2(m + 2n + 1)|\mathcal{S}|$ constraints.

2.2 Formulation of Controlled Tabular Adjustment (CTA)

Instead of suppressing cells, CTA computes an alternative safe table x: the closest to a using some particular distance $\ell_{(w)}$ based on cell weights w. In this context *safe* means that the values of sensitive cells 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:

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

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

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

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" ($z_s \le -lpl_s$). For distance ℓ_1, the resulting mixed integer linear optimization formulation is

$$\begin{aligned} &\min_{z^+, z^-} \sum_{i=1}^{n} w_i (z_i^+ + z_i^-) \\ &\text{s. to } A(z^+ - z^-) = 0 \\ &\qquad 0 \le z_i^+ \le u_{z_i} \quad i \notin \mathcal{S} \\ &\qquad 0 \le z_i^- \le -l_{z_i} \quad i \notin \mathcal{S} \\ &\qquad upl_i y_i \le z_i^+ \le u_{z_i} y_i \quad i \in \mathcal{S} \\ &\qquad lpl_i (1 - y_i) \le z_i^- \le -l_{z_i}(1 - y_i) \quad i \in \mathcal{S} \\ &\qquad y_i \in \{0, 1\} \quad i \in \mathcal{S}. \end{aligned} \tag{6}$$

where z_i, $i = 1, \ldots, n$, 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.

2.3 Formulation of Interval Protection (IP)

The purpose of IP is to replace cell values a_i by feasible intervals $[lb_i, ub_i]$, $i = 1, \ldots, n$—where feasible means that $l_i \le lb_i$ and $ub_i \le u_i$, such that estimates of

a_s, $s \in \mathcal{S}$, computed by any attacker should be outside the protection interval $[a_s - lpl_s, a_s + upl_s]$. This means—similarly to what is was done for CSP—that

$$\underline{a_s} \leq a_s - lpl_s \quad \text{and} \quad \overline{a_s} \geq a_s + upl_s, \tag{7}$$

$\underline{a_s}$ and $\overline{a_s}$ being defined as

$$\underline{a_s} = \min_x x_s \quad \text{s.to } Ax = b \quad lb_i \leq x_i \leq ub_i \; i = 1, \ldots, n \qquad \text{and} \qquad \overline{a_s} = \max_x x_s \quad \text{s.to } Ax = b \quad lb_i \leq x_i \leq ub_i \; i = 1, \ldots, n. \tag{8}$$

Like in CSP, the previous problem can be formulated as a large-scale (linear in that case, instead of mixed integer linear) optimization problem. For each sensitive cell $s \in \mathcal{S}$, two auxiliary vectors $x^{l,s} \in \mathbb{R}^n$ and $x^{u,s} \in \mathbb{R}^n$ are introduced to impose, respectively, the lower and upper protection requirement of (7). The resulting optimization problem is:

$$\begin{aligned} \min_{lb,ub} \quad & \sum_{i=1}^{n} w_i (ub_i - lb_i) \\ \text{s.to} \quad & \\ & \left.\begin{array}{rll} & Ax^{l,s} = b & \\ lb_i \leq & x_i^{l,s} \leq ub_i & i = 1, \ldots, n \\ & x_s^{l,s} \leq a_s - lpl_s & \\ & & \\ & Ax^{u,s} = b & \\ lb_i \leq & x_i^{u,s} \leq ub_i & i = 1, \ldots, n \\ & x_s^{u,s} \geq a_s + upl_s & \end{array}\right\} \quad \forall\, s \in \mathcal{S} \\ & l_i \leq lb_i \leq a_i \quad i = 1, \ldots, n \\ & a_i \leq ub_i \leq u_i \quad i = 1, \ldots, n. \end{aligned} \tag{9}$$

Problem (9) is very large, with $2n(|\mathcal{S}| + 1)$ continuous variables and $2(m + 2n + 1)|\mathcal{S}|$ constraints. On the other hand, unlike CSP, it is linear (no binary, no integer variables), and thus theoretically it can be efficiently solved in polynomial time by general or by specialized interior-point algorithms. As far as we know, no efficient implementation has been developed yet for IP, and there are only some preliminary prototypes [8]. Some related heuristics for variations of this problem were considered in [16].

3 Computational Experience

To study the effect of symmetric and asymmetric protection levels for CSP, CTA and IP we generated a set of six hierarchical tables using the generator introduced in [5]. Two versions of each table were considered: one with symmetric protection

Table 1. Instances dimensions.

Instance	n	$\|\mathcal{S}\|$	m	nnz
1	1444	135	171	2964
2	2544	237	303	5216
3	2108	196	243	4318
4	1444	216	152	2945
5	1488	220	173	3040
6	1411	209	168	2890

levels, the other with asymmetric ones. In the symmetric case a 20% of the cell value was considered, while a 5% and 20% were used for respectively the lower and upper protection levels for asymmetric instances. A priori bounds l and u were 0 and a large value, respectively (so they were always inactive). Table 1 reports the number of cells, sensitive cells, table linear relations, and number of nonzero entries of matrix A, for each instance. For CSP and CTA we used the efficient (C++) implementations described in [1,7], respectively. For IP the prototype code (a Benders decomposition implemented in AMPL [3,12]) of [8] was used.

Solution times for CSP and CTA appear in Table 2. Since the IP prototype is quite inefficient, their times are not reported. The optimality gap (i.e., the relative distance between the upper and lower bound of the optimization problem) required in both methods was 0.1%. Two symmetric tables exceeded the one hour time limit for CSP, so the final gap reached (in brackets) is notably higher than the required one. From Table 2, CTA is clearly more efficient for asymmetric than for symmetric instances; this is consistent with the results of [9], albeit they were for a quadratic version of CTA (i.e., using the ℓ_2 Euclidean instead of the ℓ_1 distance in the objective function). For CSP the pattern is not so definitive: asymmetric instances 4, 5 and, specially, 6 were slower than the corresponding symmetric variants. However, for the two largest instances (2 and 3) the symmetric cases were clearly outperformed by the asymmetric ones.

Table 2. Computation times, in seconds.

	CTA		CSP	
Instance	Symm.	Asymm.	Symm.	Asymm.
1	2.03	0.35	39.12	37.11
2	12.23	0.53	3600 (73%)	638.42
3	10.44	0.65	3600 (74%)	513.61
4	4.66	0.84	20.88	73.17
5	5.15	1.20	25.99	88.06
6	5.30	1.14	53.31	693.1

As for IP, not being a mixed integer linear problem, we do not expect differences in CPU times between symmetric and asymmetric instances (and, in addition, we would need an efficient IP code to check them, which is not the case). However we can perform a comparison between the quality of the intervals obtained for symmetric and asymmetric variants. In this respect, we first observed that most of the cells were not replaced by an interval, that is, lb_i and ub_i were the same value. Table 3 shows the number and percentage of cells which have been replaced in each instance by an interval, that is, one with different endpoints lb_i and ub_i. About 9.3% of cells are sensitive in instances 1 to 3, so one out of two interval-replaced cells is non-sensitive in the symmetric cases. Instances 4 to 6, with higher proportion of sensitive cells (about 15%), show lower rates of non-sensitive cells among all the interval-replaced cells. In all the instances, the number of cells replaced by an interval increases slightly for the asymmetric cases.

Table 3. Count and percentage of cells which have been replaced by an interval by IP.

	Symmetric		Asymmetric	
Instance	N. of cells	(%)	N. of cells	(%)
1	263	18.2	309	21.4
2	471	18.5	529	20.8
3	403	19.1	451	21.4
4	334	23.1	387	26.8
5	377	25.3	412	27.7
6	360	25.5	401	28.4

The quality of the protection is given by its difficulty to disclose the original cell values. In principle, an interval should be safe since any value inside it has the same chance to be the value sought by the attacker. However, we (somehow unexpectedly) found that an instance with symmetric protection levels is far more vulnerable. Table 4 describes the proportion of cells that have been replaced with an interval whose midpoint is exactly the original value (represented here by the zero value). The intervals have been standardized to have a width of 100. The five classes represented are given by the midpoint position: for instance, -50 means that the interval is $[-100, 0]$, that is, the rightmost value is equal to the original cell value; $(-50, 0)$ means that the original cell value is located somewhere strictly between the midpoint and the right endpoint. The proportion of cells lying in the midpoint (0) is very large among the symmetric cases, and represents a real risk of disclosure, since just taking the average of the interval has many chances to guess the original cell value. On the other hand, the proportion of such cases in instances with asymmetric protection levels is negligible. Figure 1 compares a typical instance (number 3), showing graphically the benefit of dealing with asymmetric protection levels. The other instances studied exhibited a similar behaviour.

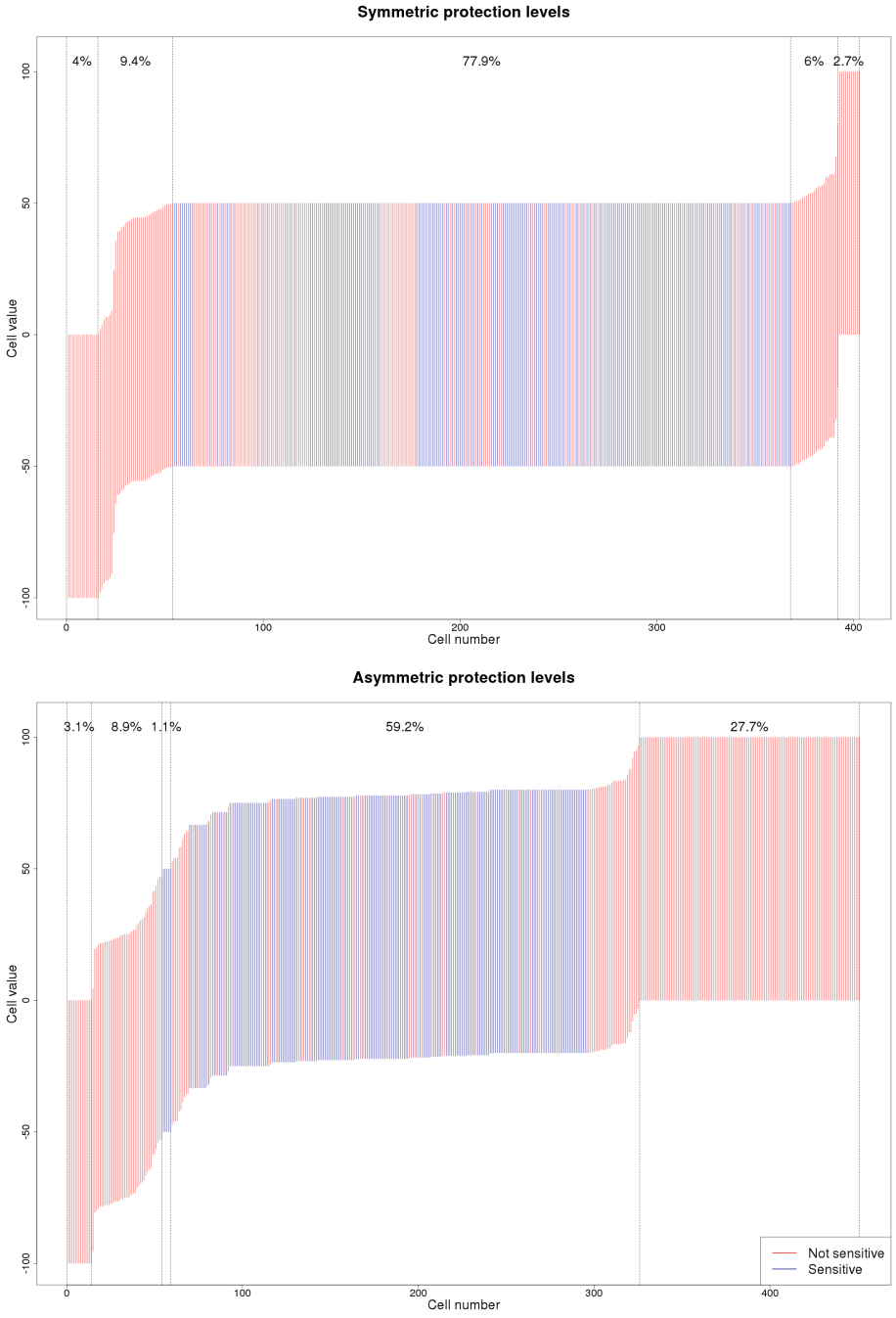

Fig. 1. Instance number 3, showing standardized intervals in both symmetric and asymmetric cases. The cells have been ranked in each case according to their interval (so the position along the x-axis is usually different in the two plots).

Table 4. Percentage distribution of intervals position.

	Instance	-50	$(-50, 0)$	0	$(0, 50)$	50
Symm.	1	2.7	5.7	87.1	4.6	0
	2	4	6.4	85.4	4	0.2
	3	4	9.4	77.9	6	2.7
	4	4.2	6.6	80.8	6.6	1.8
	5	6.4	7.4	77.5	6.9	1.9
	6	4.7	8.1	76.9	8.3	1.7
Asymm.	1	5.8	3.9	2.9	60.8	26.2
	2	2.3	4.5	2.3	59.2	31.4
	3	3.1	8.9	1.1	59.2	27.7
	4	1.6	4.4	2.1	68.7	23
	5	3.6	5.8	1	67.7	21.6
	6	2	6	3.5	67.3	20.9

4 Conclusions

From the computational results in the solution of a set of six hierarchical tables, using efficient implementations of CSP and CTA, and a prototype code for IP, we conclude:

- For the mixed integer linear problems CTA and CSP, symmetry of protection levels has an impact on the solution time. For CTA this assertion was always true: asymmetric instances were faster than symmetric ones. For CSP this fact was not so conclusive: only for the largest instances tested asymmetry provided faster executions.
- For IP asymmetric protection levels affected the quality of the solution, rather than solution times. In general, symmetric protection levels provided very poor intervals, and in most cases their midpoints disclosed the true cell value. Therefore, the use of asymmetric protection levels in IP should be highly recommended.
- Protection levels automatically provided by sensitivity rules are always symmetric: this practice should be reconsidered according to the results of this work.

A similar analysis to that done for IP could be performed for the intervals obtained by the auditing phase of the CSP; this is part of the future work to be done.

References

1. Baena, D., Castro, J., Frangioni, A.: Stabilized Benders methods for large-scale combinatorial optimization, with application to data privacy. Research report DR 2017/03, Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Catalonia (2017)
2. Baena, D., Castro, J., González, J.A.: Fix-and-relax approaches for controlled tabular adjustment. Comput. Oper. Res. **58**, 41–52 (2015)
3. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Comput. Manage. Sci. **2**, 3–19 (2005). English translation of the original paper appeared in Numerische Mathematik, 4 (1962) 238–252
4. Castro, J.: Minimum-distance controlled perturbation methods for large-scale tabular data protection. Eur. J. Oper. Res. **171**, 39–52 (2006)
5. Castro, J.: A shortest paths heuristic for statistical disclosure control in positive tables. INFORMS J. Comput. **19**, 520–533 (2007)
6. Castro, J.: Recent advances in optimization techniques for statistical tabular data protection. Eur. J. Oper. Res. **216**, 257–269 (2012)
7. Castro, J., González, J.A., Baena, D.: User's and programmer's manual of the RCTA package. Technical report DR 2009/01, Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Catalonia (2009)
8. Castro, J., Via, A.: Revisiting interval protection, a.k.a. partial cell suppression, for tabular data. In: Domingo-Ferrer, J., Pejić-Bach, M. (eds.) PSD 2016. LNCS, vol. 9867, pp. 3–14. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45381-1_1
9. Castro, J., Frangioni, A., Gentile, C.: Perspective reformulations of the CTA problem with L_2 distances. Oper. Res. **62**, 891–909 (2014)
10. Fischetti, M., Salazar, J.J.: Solving the cell suppression problem on tabular data with linear constraints. Manage. Sci. **47**, 1008–1026 (2001)
11. Fischetti, M., Salazar, J.J.: Partial cell suppression: a new methodology for statistical disclosure control. Stat. Comput. **13**, 13–21 (2003)
12. Fourer, R., Gay, D.M., Kernighan, D.W.: AMPL: A Modeling Language for Mathematical Programming. Duxbury Press, Duxbury (2002)
13. González, J.A., Castro, J.: A heuristic block coordinate descent approach for controlled tabular adjustment. Comput. Oper. Res. **38**, 1826–1835 (2011)
14. Hundepool, A., et al.: Statistical Disclosure Control. Wiley, Chichester (2012)
15. Kelly, J.P., Golden, B.L., Assad, A.A.: Cell suppression: disclosure protection for sensitive tabular data. Networks **22**, 28–55 (1992)
16. Robertson, D.: Automated disclosure control at Statistics Canada. In: Paper presented at the second international seminar on statistical confidentiality, Luxembourg (1994)
17. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM, Philadelphia (1997)