ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 171 (2006) 39–52

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Stochastics and Statistics

Minimum-distance controlled perturbation methods for large-scale tabular data protection

Jordi Castro *

Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Pau Gargallo 5, 08028 Barcelona, Spain

Received 3 July 2003; accepted 9 August 2004
Available online 2 November 2004

Abstract

National Statistical Agencies routinely release large amounts of tabular information. Prior to dissemination, tabular data needs to be processed to avoid the disclosure of individual confidential information. One widely used class of methods is based on the modification of the table cells values. However, previous approaches were not able to preserve the values of the marginal cells and the additivity relations for a general table of any dimension, size and structure. This void was recently filled by the controlled tabular adjustment and one of its variants, the quadratic minimum-distance controlled perturbation method. Although independently developed, both approaches rely on the same strategy: given a set of tables to be protected, they find the minimum-distance values to the original cells that make the released information safe. Controlled tabular adjustment uses the L_1 distance; the quadratic minimum-distance variant considers L_2. This work presents both approaches within an unified framework, and includes a new variant based on L_∞. Among other benefits, the unified framework permits the simple comparison of the three distances, and a single general result about their disclosure risk. The three distances are evaluated with the unique standard library for tabular data protection currently available. Some of the complex instances were contributed by National Statistical Agencies, and, therefore, are good representatives of theirs real needs. Unlike alternative methods, the three distances were able to solve all the instances, requiring only few seconds for each of them on a personal computer using a general purpose solver. The results show that this class of methods are an effective and promising tool for the protection of large volumes of tabular data. All the linear and quadratic problems solved in the paper are delivered to the optimization community in MPS format.

Keywords: Statistical confidentiality; Statistical disclosure control; Controlled tabular adjustment, Linear programming; Quadratic programming; Interior-point methods

* Tel.: +34 93 4015854; fax: +34 93 4015855.
E-mail address: jcastro@eio.upc.es

0377-2217/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2004.08.034

1. Introduction

The safe dissemination of data is one of the main concerns of National Statistical Agencies. The released data can be classified as disaggregated or aggregated. Disaggregated data (a.k.a. microdata or microfiles) consists of files of records, each record providing the values for a set of variables of an individual. Aggregated data (a.k.a. tabular data) is obtained from microdata crossing two or more variables, which results in sets of tables with a likely large number of cells. It must be guaranteed, for both types of data, that no individual information can be derived from the released information. The available methods for this purpose belong to the field of statistical disclosure control. Good introductions to the state-of-the-art in this field can be found in the monographs Willenborg and de Waal (2000) and Domingo-Ferrer (2002).

In this paper we focus on tabular data protection. Although each cell of the table shows aggregated information for several individuals, there is a risk of disclosing individual data. This is clearly shown in the example of Fig. 1. Table (a) of that figure gives the average salary for age interval and ZIP code, while table (b) shows the number of individuals for the same variables. If there was only one individual in ZIP code z_2 and age interval 51–55, then any external attacker would know the salary of this single person is 40,000€. For two individuals, any of them could deduce the salary of the other, becoming an internal attacker. Usually, cells showing information about few individuals are considered sensitive, although other rules can be used in practice. Methods for detecting sensitive cells are out of the scope of this work. A recent discussion about sensitivity rules can be found in Domingo-Ferrer and Torra (2002), and Robertson and Ethier (2002).

		z_1	z_2	
⋮	...	...	...	...
51–55	...	38000€	40000€	...
56–60	...	39000€	42000€	...
⋮	...	...	...	...

(a)

		z_1	z_2	
⋮	...	...	...	...
51–55	...	20	1 or 2	...
56–60	...	30	35	...
⋮	...	...	...	...

(b)

Fig. 1. Example of disclosure in tabular data: (a) average salary per age and ZIP code, (b) number of individuals per age and ZIP code. If there is only one individual in ZIP code z_2 and age interval 51–55, then any external attacker knows the salary of this single person is 40,000€. For two individuals, any of them can deduce the salary of the other, becoming an internal attacker.

Fig. 1 shows a two-dimensional example. This can be considered the simplest case. However, in practice we must deal with more complex situations, including multidimensional, hierarchical and linked tables. Multidimensional tables are obtained crossing more than two variables, and they can be individually protected. Hierarchical tables are sets of tables whose variables have a hierarchical relation (e.g., ZIP code and city). In that case, the total or marginal cells of some tables are internal ones for the others. They have to be protected together, to avoid the disclosure of sensitive data. Finally, linked tables are a generalization of the previous situation, where several tables are made from the same microdata, thus sharing information or cells, either hierarchical or not. Again, they have to be protected together. Linked tables can deal with any table dimension, size and structure, and thus include the other situations. Dealing with linked tables is a desired feature of any tabular protection method. Eventually, the final goal would be the protection of the whole set of linked tables that can be produced from some microfiles (e.g., a population census). Clearly, the number of cells involved in that case might be of several millions, an impractical size for most current tabular protection techniques. The family of protection methods considered in this work deal with linked tables, and, as shown in the computational results, can solve real-world large instances in few seconds. All the above situations can both refer to frequency tables (i.e., cell values are integer and are usually associated to the number of individuals in that cell) or magnitude tables (i.e., cell values are real, and, for instance, they show the mean for some other variable of all the individuals in that cell). In this work we focus on tables of magnitudes. For tables of frequencies the procedures here described can also be applied followed by some heuristic post-process.

Current methods for tabular data protection can be classified as perturbative (they change the cell values) or nonperturbative (no change is performed). The most widely used nonperturbative method is *cell suppression*, where some *secondary* cells are removed to avoid the disclosure of some sensitive *primary* cells (which are removed as well). That results in a difficult combinatorial optimization problem, which finds the pattern of secondary suppressions that makes the table safe with a minimum number of cells or information loss. Some heuristics for two and three-dimensional tables (Kelly et al., 1992; Carvalho et al., 1994; Cox, 1995; Dellaert and Luijten, 1999; Castro, 2002b, 2004a) and exact methods for two-dimensional and general linked tables (Fischetti and Salazar, 1999, 2001) have been suggested for the cell suppression problem. The main inconvenience of this approach is that, due to its combinatorial nature, the solution of very large instances (with possibly millions of cells) can result in impractical execution times.

Among the perturbative approaches, one of the techniques that received more attention was *rounding*. This method rounds cell values to a multiple of a fixed integer rounding base. *Controlled rounding* is a variant where the additivity of the table is preserved (i.e., rounded marginal values are the sum of the corresponding slice of internal rounded cells). Initially introduced in Bacharach (1966), efficient methods could only be developed for two-dimensional tables (Cox and Ernst, 1982), possibly with subtotals (Cox and George, 1989). For three-dimensional tables controlled rounding is a NP-hard problem (Kelly et al., 1990a). Several heuristics (Kelly et al., 1990b) and exact approaches (Kelly et al., 1990c) were devised, but were only applied to small size tables. The NP-hardness of the approach makes it impractical for large tables, as the real-world ones tested in this work. Moreover, in practice it can be necessary to maintain some (possibly all) of the original total cells, instead of rounding them.

To avoid the above lacks of rounding, Dandekar and Cox (2002) first introduced the controlled tabular adjustment method (CTA). Extensions to CTA have been recently considered for univariate and multivariate statistics (Cox et al., 2004). Independently, Castro (2002a) suggested a similar controlled perturbation method with a quadratic objective function. The resulting quadratic optimization problem is efficiently solved by an interior-point method in Castro (2004b). Both approaches find the minimum-distance (or closest) tables to those to be protected, preserving marginal values, if required, as well as any set of additional linear constraints. That means we try to minimize the information loss when delivering the perturbed values. CTA and the quadratic minimum-distance controlled perturbation are essentially the same method. The main difference is the distance considered in the objective function: CTA uses L_1 while the quadratic minimum-distance controlled perturbation method uses L_2. The main contribution of the paper is that it presents both distances within an unified framework, and includes a new variant based on L_∞. Some of the benefits of presenting those distances under an unified framework are: (1) They can be easily compared, as done in the paper. (2) A single general result about the disclosure risk of the distances can be developed independently of the objective function of the final formulation; this is also done in the paper. (3) It is possible to combine the different distances in one single objective function (Castro, 2004c).

The main features of CTA or minimum-distance controlled perturbation methods are:

- Efficient: we will show that real-world large instances can be solved in few seconds using current linear and quadratic programming technology.
- Versatile: they deal with any table or set of tables, and with any additional linear constraint (e.g., preserving the value of total cells).
- Safe: as it will be shown, even with partial information, an attacker is not able to reproduce the original data.
- Simple: they have a straightforward derivation and formulation. That is a very appreciated feature by National Statistical Agencies' staff, which tend to avoid methods based on sophisticated procedures (Dandekar, 2003b).

Alternative approaches for tabular data protection have flaws in some of the above features.

The present work deals with methods for achieving controlled perturbation while optimizing specific measures of data quality (namely, minimum distance or change from the original table as measured by particular L_p norms). Other approaches (Cox et al., 2004) deal with achieving controlled tabular adjustment while controlling change to statistical properties of the original table (means, variances, etc.) and between tables (covariances, correlations, regression). Both notions of data quality are important and appropriate in various contexts.

Recently, Dandekar (2003a) introduced an alternative perturbation approach, computationally more efficient that the family of methods here considered. However, such approach cannot preserve the value of total cells, which is a desirable property in practice (rather, total cells suffer the largest perturbations).

The structure of the document is as follows. Section 2 describes the *CTA* or *minimum-distance controlled perturbation* framework. Sections 2.1, 2.2 and 2.3 detail the variants associated to L_1, L_2 and L_∞, respectively. Section 3 compares the optimization problems derived from these three particular distances. Section 4 analyzes the disclosure risk of the method, showing it is safe. Finally, Section 5 presents some computational results in the solution of some real-world large instances. These computational results show the effectiveness of the approach.

2. The CTA or minimum-distance controlled perturbation framework

This section describes the general model, and the particular formulations for the L_1, L_2 and L_∞ distances. More details can be found in Dandekar and Cox (2002), and Cox et al. (2004).

Any table or list of tables, of any dimension, size and structure, can be represented as an array of cells a_i, $i = 1,\ldots,n$, that satisfy a set of m linear relations

$$Ma = b, \tag{1}$$

$a \in \mathbb{R}^n$ being the vector of a_i's, $b \in \mathbb{R}^m$ the right-hand-side term of the linear relations, and $M \in \mathbb{R}^{m\times n}$ the cell relations matrix. In practice most tables have positive cell values, and constraints

$$a \geqslant 0 \tag{2}$$

must be added to (1).

Given a set $\mathscr{P}$ of indices of sensitive or confidential cells, the *controlled tabular adjustment* or *minimum-distance controlled perturbation* method finds, according to some metric, the closest values x_i to a_i, $i = 1,\ldots,n$, that satisfy the table relations (1) and, if needed, (2), such that $x_i, i \in \mathscr{P}$—the values of the sensitive cells—are safe (safety is discussed below). This model can be applied to any kind of table or set of tables, since it does not constraint the structure of the cell relations $Ma = b$. Any other set of linear relations can also be included to this model.

This general model can be formulated as

$$\text{(P1)}\quad \begin{array}{lll} \min\limits_{x} & \|x - a\|_L & (3)\\ \text{subject to} & Mx = b, & (4)\\ & l_x \leqslant x \leqslant u_x, & (5) \end{array}$$

$x \in \mathbb{R}^n$ being the vector of perturbed cell values. L in (3) denotes the distance to be used, which can be affected by any positive semidefinite diagonal metric matrix $W = \text{diag}(w_1,\ldots,w_n)$. In the computational results of Section 5 we used $w_i = 1/a_i$, if $a_i \neq 0$, otherwise $w_i = 1$. The three more reasonable choices for L are L_1, L_2 and L_∞. They are discussed in the following sections. (4) guarantees x is a well-formed table. The bounds (5) are used to deal with the level of knowledge any attacker has about the cell values, and to guarantee the safety of the perturbed table, as follows:

- We assume any attacker knows a lower and upper bound, respectively $\underline{a}_i$ and $\overline{a}_i$, for each cell a_i, $i = 1,\ldots,n$. If no previous knowledge is assumed for cell i, we simply set $\underline{a}_i = 0$ ($\underline{a}_i = -\infty$ if bounds (2) were omitted) and $\overline{a}_i = +\infty$. (5) includes bounds $\underline{a}_i \leqslant x_i \leqslant \overline{a}_i$.
- The protection of each sensitive cell $i \in \mathscr{P}$ is achieved through a lower and upper protection levels, respectively lpl_i and upl_i, such that the released value should be greater or equal than $a_i + upl_i$ or less or equal than $a_i - lpl_i$. These protection levels are provided by the user (e.g., the National Statistical Agency), and they are usually a fraction of the cell value a_i. We assume

that the user fixes in advance the sense of the protection for each sensitive cell, i.e., if the cell will be protected by its upper or its lower protection level. Therefore, (5) includes one of the bounds $x_i \geqslant a_i + upl_i$ or $x_i \leqslant a_i - lpl_i$.

If the values of a large number of cells want to be preserved, problem (P1) can be infeasible. This can happen, e.g., for small instances if marginal cells are maintained in the perturbed table. For large tables, infeasibility should rarely occur. However, if needed, we can replace in (5) the bounds $a_i \leqslant x_i \leqslant a_i$ of the cells fixed to the original value by the penalization $P\|x_i - a_i\|_L$ in the objective function, P being a large penalty parameter.

If, instead of being a user decision, we want the mathematical programming problem (P1) to choose the best sense for sensitive cells, either $x_i \geqslant a_i + upl_i$ or $x_i \leqslant a_i - lpl_i$, we need a binary variable and two extra constraints for each of them:

$$\begin{aligned} & x_i \geqslant -S(1-y_i) + (a_i + upl_i)y_i, \quad i \in \mathscr{P}, \\ & x_i \leqslant Sy_i + (a_i - lpl_i)(1-y_i), \quad i \in \mathscr{P}, \\ & y_i \in \{0,1\}, \quad i \in \mathscr{P}. \end{aligned} \tag{6}$$

S in (6) is a large value (e.g., $S = \sum_{i=1}^{n} a_i$). When $y_i = 1$, constraints (6) imply $S \geqslant x_i \geqslant (a_i + upl_i)$. When $y_i = 0$ we have $-S \leqslant x_i \leqslant (a_i - lpl_i)$. That results in a large combinatorial optimization problem, which would constraint the effectiveness of the approach to small and medium sized problems. Moreover, in practice tabular data protection is the last stage of the "data cycle", and, in an attempt to meet publication deadlines, National Statistical Agencies require fast solutions to large and complex tables (Dandekar, 2003b). Therefore, instead of solving the combinatorial optimization problem, we can heuristically decide in advance the sense for each sensitive cell ($y_i = 1$ or $y_i = 0$) and then solving the optimization problem (P1). That solution will be an upper bound for the combinatorial optimization problem. Some straightforward heuristics were suggested in Dandekar and Cox (2002), but, from the reported computational experience, they provided similar results. The particular choice of y_i values do not affect the safety of the released perturbed table, but only the deviations from the original cell values.

The general problem (P1) can also be formulated in terms of deviations or perturbations from the current cell values. Indeed, defining

$$x_i = a_i + z_i, \quad i = 1, \ldots, n, \tag{7}$$

the optimization problem (P1) can be transformed to

(P2)

$$\min_z \quad \|z\|_L \tag{8}$$

$$\text{subject to} \quad Mz = 0, \tag{9}$$

$$l_z \leqslant z \leqslant u_z, \tag{10}$$

where $z \in \mathbb{R}^n$ is the vector of deviations, and

$$l_z = l_x - a, \qquad u_z = u_x - a. \tag{11}$$

Two benefits of the formulation in terms of deviations are:

- The cell values a_i of the real table are not needed to solve the optimization problem (P2). Only the cell relations and deviations bounds, represented by matrix M and vectors l_z and u_z, are required. Therefore, the solution of the above optimization problem can be performed by an external entity (e.g., if some nonavailable software or hardware was required) without delivering the original cell values.
- Two tables with the same cell relations and bounds, that only differ in the cell values (e.g., corresponding to data of two different years or census), are protected with the same perturbations. Therefore, the optimization problem (P2) only needs to be solved once.

Next three subsections specialize the general model for the L_1, L_2, and L_∞ distances, using the formulation in terms of deviations.

2.1. The L_1 objective

Using the L_1 distance, problem (P2) becomes

(P3)

$$\min_z \quad \sum_{i=1}^{n} w_i |z_i| \tag{12}$$

subject to (9), (10).

To transform the above into an equivalent linear programming problem, we replace each z_i by the difference of two nonnegative variables, z_i^+ and z_i^-, associated respectively with the positive and negative deviations:

$$z_i = z_i^+ - z_i^-, \quad i = 1, \ldots, n. \tag{13}$$

The resulting linear programming problem is

(P4)

$$\begin{aligned} \min_{z^+,z^-} \quad & \sum_{i=1}^{n} w_i(z_i^+ + z_i^-) && (14)\\ \text{subject to} \quad & M(z^+ - z^-) = 0, && (15)\\ & l_z \leqslant z^+ - z^- \leqslant u_z, && (16)\\ & z^+ \geqslant 0, \quad z^- \geqslant 0, && (17) \end{aligned}$$

$z^+ \in \mathbb{R}^n$ and $z^- \in \mathbb{R}^n$ being respectively the vectors of positive and negative deviations.

Eqs. (16) and (17) can be simplified. For a nonsensitive cell i, l_{z_i} and u_{z_i}, as defined in (11), will respectively be negative and positive. Then, for nonsensitive cells, Eqs. (16) and (17) reduce to

$$\begin{aligned} & 0 \leqslant z_i^+ \leqslant u_{x_i} - a_i, \quad i \notin \mathscr{P},\\ & 0 \leqslant z_i^- \leqslant a_i - l_{x_i}, \quad i \notin \mathscr{P}. \end{aligned} \tag{18}$$

For a sensitive cell i, the equations to be used depend on the sense of the protection considered, defined in (6) by the binary variable y_i. If the sense is "upper" (i.e., $y_i = 1$) then we must impose

$$\begin{aligned} & upl_i \leqslant z_i^+ \leqslant u_{x_i} - a_i, \quad i \in \mathscr{P}, \ y_i = 1,\\ & z_i^- = 0, \quad i \in \mathscr{P}, \ y_i = 1. \end{aligned} \tag{19}$$

If the sense is "lower" (i.e., $y_i = 0$) then we need

$$\begin{aligned} & z_i^+ = 0, \quad i \in \mathscr{P}, \ y_i = 0,\\ & lpl_i \leqslant z_i^- \leqslant a_i - l_{x_i}, \quad i \in \mathscr{P}, \ y_i = 0. \end{aligned} \tag{20}$$

The final linear programming problem to be solved is

(P5)

$$\begin{aligned} \min_{z^+,z^-} \quad & (14)\\ \text{subject to} \quad & (15), (18), (19), (20). \end{aligned} \tag{21}$$

Using $w_i = 1/a_i$ if $a_i \neq 0$, as in the computational results of Section 5, the objective function to be minimized is the total relative deviation between the original and the perturbed data. Problem (P5) is basically the same model as that of Dandekar and Cox (2002), there obtained with a different derivation.

2.2. The L_2 objective

Using the L_2 distance, and removing the square root of the objective, problem (P2) becomes

(P6)

$$\begin{aligned} \min_{z} \quad & \sum_{i=1}^{n} w_i z_i^2 \\ \text{subject to} \quad & (9), (10). \end{aligned} \tag{22}$$

Using $w_i = 1/a_i$ if $a_i \neq 0$, as in the computational results of Section 5, the objective function corresponds to the χ^2 distance between the original and the perturbed data (Cox, 2003).

2.3. The L_∞ objective

In this case, problem (P2) is

(P7)

$$\begin{aligned} \min_{z} \quad & \max_{i=1\ldots n}\{w_i|z_i|\}\\ \text{subject to} \quad & (9), (10). \end{aligned}$$

To remove absolute values, we proceed as in Section 2.1, replacing each variable by the difference of two positive variables. Moreover, it seems reasonable to consider separately the deviations for the sensitive and nonsensitive cells, since the former are forced to be greater than zero whereas the latter should be as close as possible to zero. The problem to be solved is thus

(P8)

$$\begin{aligned} \min_{z^+,z^-} \quad & \left(\max_{i\in\mathscr{P}}\{w_i(z_i^+ + z_i^-)\} + \max_{i\notin\mathscr{P}}\{w_i(z_i^+ + z_i^-)\} \right)\\ \text{subject to} \quad & (15), (18), (19), (20). \end{aligned}$$

To transform the above into a linear programming problem we add two extra variables, $z_{\in P}$ and $z_{\notin P}$, which will store the maximum deviation for, respectively, the sensitive and nonsensitive cells. The equivalent linear programming problem can be written as

(P9)

$$\begin{aligned} \min_{z^+,z^-,z_{\in\mathscr{P}},z_{\notin\mathscr{P}}} \quad & z_{\in\mathscr{P}} + z_{\notin\mathscr{P}} \\ \text{subject to} \quad & (15), (18), (19), (20), \\ & z_{\in\mathscr{P}} \geqslant w_i(z_i^+ + z_i^-), \quad i \in \mathscr{P}, \\ & z_{\notin\mathscr{P}} \geqslant w_i(z_i^+ + z_i^-), \quad i \notin \mathscr{P}. \end{aligned} \tag{23}$$

3. Comparison of the three optimization problems

The distances of Sections 2.1–2.3 gave rise to three different optimization problems, whose main features are shown in Table 1. Only the most efficient solution algorithms for the type of problem are reported. The L_2 objective provides the smallest problem, but it can only be efficiently solved by an interior-point algorithm (Wright, 1997). For the other two problems we can either use an interior-point algorithm or the simplex method (Dantzig, 1963). The efficiency of those methods depends on the particular structure of the problem (Bixby, 2002), and, as it will shown in Section 5, it is difficult to know in advance which will be the fastest option for a particular instance. A theoretical advantage of interior-point algorithms is that they have a polynomial complexity, both for linear and quadratic optimization problems. On the other hand, although it is a nonpolynomial algorithm, recent developments in the dual simplex made it a very effective approach (Bixby, 2002). It is worth to note that the computational cost for the quadratic problem (P6), solved through an interior-point algorithm, is the same as if it was linear, because it has a separable objective function (i.e., there are no products of two different variables) (Wright, 1997). Moreover, in the tabular data protection context, interior-point algorithms can be specialized to efficiently solve very large instances (Castro, 2000, 2004b).

In some cases, for the L_2 distance, we can obtain a closed-form solution. For instance, if we fix the deviations of sensitive cells (i.e., $z_i = upl_i$ or $z_i = -lpl_i$) and remove the inequality constraints (i.e., either we assume they are inactive or we can accept negative values in the perturbed table), problem (P6) can be rewritten as

(P10)

$$\begin{aligned} \min_{z} \quad & \sum_{i=1}^{n} z^{\mathrm{T}} W z \\ \text{subject to} \quad & Az = b, \end{aligned} \tag{24}$$

where $Az = b$ include the original constraints $Mz = 0$ and those that fix the values of the deviations of sensitive cells. The solution of problem (P10) has the following closed form:

$$z^* = W^{-1}A^{\mathrm{T}}(AW^{-1}A^{\mathrm{T}})^{-1}b. \tag{25}$$

The computational effort of (25) and that of an iteration of a quadratic interior-point algorithm are the same (Wright, 1997). That means that, if inequalities are inactive at the optimum, the interior-point algorithm will perform several iterations, when the solution of (25) would suffice. This wasted effort can be avoided by computing the initial values of the interior-point algorithm through (25). If that initial point satisfies the inequalities, then we already have the solution. Otherwise we start the iterative steps. That simple strategy permits to accommodate the interior-point algorithm with no extra effort to both instances with and without active constraints.

Fig. 2 shows the consequences of the three distances on a small example. The original data a to

Table 1
Properties of the three optimization problems

	L_1, problem (P5)	L_2, problem (P6)	L_∞, problem (P9)
Number of variables	$2n$	n	$2n+2$
Number of constraints	m	m	$m+n$
Type of problem	Linear	Quadratic	Linear
Solution algorithms	Simplex and interior-point	Interior-point	Simplex and interior-point

a

10$_{(3)}$	15	11	9	45
8	10	**12**$_{(4)}$	15	45
10	12	**11**$_{(2)}$	**13**$_{(5)}$	46
28	37	34	37	136

(a)

x_{L_1}

13	18	5	9	45
8	11	16	10	45
7	8	13	18	46
28	37	34	37	136

(b)

x_{L_2}

13	18.627	5	8.373	45
8.173	10.200	16	10.627	45
6.827	8.173	13	18	46
28	37	34	37	136

(c)

x_{L_∞}

13	22	5	5	45
$10.\widehat{45}$	$4.\widehat{54}$	16	14	45
$4.\widehat{54}$	$10.\widehat{45}$	13	18	46
28	37	34	37	136

(d)

Fig. 2. Behaviour of the three objective functions on a small example table (a) original data a to be protected. Sensitive cells are in boldface, and upper protection levels are given in brackets. The upper protection sense was considered for the four sensitive cells. (b), (c), (d) respectively, optimal perturbed data x_{L_1}, x_{L_2} and x_{L_∞} obtained with each objective. For the three objectives we used weights $w_i = 1/a_i$, inactive bounds $\underline{a_i} = 0$ and $\overline{a_i} = \infty$ for all the internal cells, and marginal cells were fixed.

be protected are in Panel (a). Sensitive cells appear in boldface, and their upper protection levels upl_i are given in brackets. The upper protection sense was considered for all the sensitive cells. Panels (b), (c) and (d) of Fig. 2 show the optimal perturbed tables obtained for respectively the L_1, L_2 and L_∞ distances. For the three distances the marginal cells were fixed, and weights $w_i = 1/a_i$ and bounds $\underline{a_i} = 0$ and $\overline{a_i} = \infty$ were considered for all the internal cells. The average percentage relative deviation between the perturbed and the original data is 15.06%, 15.13% and 21.25% for respectively L_1, L_2 and L_∞. The 2-norm distance between the perturbed and the original data is 12.25, 12.14 and 14.96 for respectively L_1, L_2 and L_∞. The maximum relative percentage deviation between the perturbed and the original data is 54.5%, 54.5% and 54.5% for respectively L_1, L_2 and L_∞ (associated to cell (1,3) in the three cases). In that example, the above indicators are similar for L_1 and L_2, while L_∞ provides worse results. L_∞ even does not provide a better maximum relative deviation, which is the objective function it considers. A similar behaviour will be observed in the computational results of Section 5. It is noteworthy that L_1 provided a perturbed integer table. Although the optimization problem (P5) does not guarantee an integer solution, we observed that in most cases such property is satisfied. For frequency tables, that can be an advantage of L_1 compared to L_2.

4. Disclosure risk of the method

To retrieve the original cell values a_i from the released ones x_i, an attacker needs the applied deviations z_i. Those deviations are the solution of the optimization problem (P2). Detailing the expression for the bounds (10), the attacker should then solve

(P11)

$$\min_z \quad \|z\|_L \tag{26}$$

$$\text{subject to} \quad Mz = 0, \tag{27}$$

$$z_i \geqslant \underline{a_i} - a_i, \quad i = 1, \ldots, n, \tag{28}$$

$$z_i \leqslant \overline{a_i} - a_i, \quad i = 1, \ldots, n, \tag{29}$$

$$z_i \leqslant -lpl_i \quad \text{or} \quad z_i \geqslant upl_i, \quad i \in \mathscr{P}. \tag{30}$$

The information required for the solution of problem (P11) is:

- The particular distance L used in (26) to compute the deviations. Without this information the attacker should try to solve the problems for L_1, L_2 and L_∞, considering that one of the three solutions gives the required deviations.
- The weights w_i, $i = 1,\ldots,n$, used in (26). If $w_i = 1/a_i$, the weights are clearly unknown to the attacker.
- The constraints matrix M of (27). The attacker knows it from the cell relations of the released table.
- The lower and upper bounds $\underline{a_i} - a_i$ and $\overline{a_i} - a_i$, $i = 1,\ldots,n$, of (28) and (29), respectively. $\underline{a_i}$ and $\overline{a_i}$ are the cell value bounds that were assumed known by the attacker when protecting the original table. It can be a strong assumption to consider the attacker knows those exact values. Moreover, the original cell values a_i are clearly unknown to the attacker. However, to correctly solve problem (P11) the attacker only needs the same values for the active bounds. For nonactive bounds it is enough to use values that provide a

feasible region larger or equal than for the original problem. For instance, if the attacker guesses that all the bounds resulted inactive when protecting the table, constraints (28) and (29) can be removed. That would be the case if large bounds $\underline{a_i}$ and $\overline{a_i}$ are used by default when protecting tables (e.g., $\underline{a_i} = 0$ and $\overline{a_i} = +\infty$).

- The set $\mathscr{P}$ of sensitive cells of (30). Unlike other protection methods—as cell suppression—, the released table gives no information about which cells are sensitive, or candidates to be sensitive. Therefore, the attacker is forced to deduce sensitive cells from his/her own knowledge.
- The lower and upper protection levels lpl_i and upl_i, $i \in \mathscr{P}$, and the sense ("upper" or "lower") used in (30) for each sensitive cell when protecting the original table. In practice, that information will not be distributed with the released table. Protection levels are usually a percentage of the cell values a_i, which are unknown to the attacker. The number of variations for the protection senses is $2^{|\mathscr{P}|}$. If the senses were, for instance, randomly chosen, the attacker would be unable to reproduce them.

Except for the constraints matrix M, the rest of required terms are unknown or uncertain to the attacker. Therefore, problem (P11) cannot be solved, and the released table will be safe. However, we will analyze two unfavorable situations, where the attacker has respectively partial and complete information about the problem. Although fairly improbable in practice, they are considered to stress the low disclosure risk of the method.

4.1. Attacker with partial information

First, consider the attacker knows L, w_i, that bounds (28) and (29) are inactive—thus can be removed—, the set $\mathscr{P}$ of sensitive cells, and the sense ("upper" or "lower") of each sensitive cell. Without loss of generality, and to simplify the exposition, assume all the senses are "upper". With that information, the safety of the deviations relies on the protection levels upl_i of the sensitive cells. If the attacker can obtain approximate values $upl'_i = upl_i + e_i$, $e_i \in \mathbb{R}$, $i \in \mathscr{P}$, the problem to be solved to disclose the deviations is

$$\text{(P12)} \quad \begin{array}{ll} \min\limits_{z'} & \|z'\|_L \\ \text{subject to} & Mz' = 0, \\ & z'_i \geqslant upl_i + e_i, \quad i \in \mathscr{P}. \end{array} \tag{31}$$

If $e_i = 0$ for all $i \in \mathscr{P}$, the solution of problem (P12) can provide the deviations used to protect the table. The safety of the table thus depends on how sensitive the solution z'^* is to possible small e_i values. The relation between both terms is given by the *Lagrange multipliers* of the inequality constraints of problem (P12):

Proposition 1. *If $z'^*(e) \in \mathbb{R}^n$ is the solution of problem (P12) for a particular vector of $e = (e_1, \ldots, e_{|\mathscr{P}|})$ values, and $\mu \in \mathbb{R}^{|\mathscr{P}|}$ is the Lagrange multipliers vector of the inequality constraints of problem (P12) for $e = 0$ (i.e., the multipliers obtained when protecting the table), then*

$$\nabla_e \|z'^*(e)\|_L|_{e=0} = \mu. \tag{32}$$

Proof. This is an immediate result of the *sensitivity theorem* of optimization (see, e.g, Luenberger (1989, pp. 312–318)). □

Although not made explicit, the above proposition applies to problem (P12) once formulated as one of the optimization problems (P5), (P6) or (P9). In problems (P5) and (P9) the variables were z^+ and z^-. In that case, since we are assuming an upper sense for all the sensitive cells, only the Lagrange multipliers of the bounds $z_i^+ \geqslant upl_i$ should be considered. Moreover, for, respectively, the L_1 and L_∞ distances, problems (P5) and (P9) were linear, and, for small enough vectors $e = (e_1, \ldots, e_{|\mathscr{P}|})$, it is well-known that the relation (32) can be recast as

$$\|z'^*(e)\|_L - \|z^*\|_L = \sum_{i \in \mathscr{P}} \mu_i e_i, \tag{33}$$

z^* being the deviations used to protect the table. If the attacker does not know the set $\mathscr{P}$ of sensitive cells, and uses and approximate one $\mathscr{P}'$, the multipliers of cells $i \in \mathscr{P}' \setminus \mathscr{P}$ will also intervene in (32), decreasing even more the disclosure risk. Proposition 1 gives an indicator of the quality of the protection: tables with nonsmall Lagrange multipliers for the bounds of deviations are unlikely to be disclosed, even if the attacker has a good knowledge about the original data.

4.2. Attacker with complete information

The attacker may not be able to reproduce the right perturbations through problem (P11) even with complete information:

Proposition 2. *Assume the attacker knows all the terms of problem* (P11)*. If the L_2 distance is used, the solution of that problem will provide the deviations used to protect the table. However, for L_1 or L_∞, the attacker can obtain alternative deviations.*

Proof. The objective function of problem (P6), for the L_2 distance, is strictly convex, and thus has a unique minimizer on the feasible region. For L_1 and L_∞, the objective functions of problems (P5) and (P9) are linear, and different algorithms or implementations can provide alternative solutions. □

Indeed, we observed that, in practice, two implementations of the simplex method provided very different deviations patterns for L_1.

With the above information and that of Section 3 we can discuss the theoretical benefits and disadvantages of the three distances before the computational experience of next section. As shown before, the minimum-distance controlled perturbation method has a low disclosure risk with any of the three distances. Proposition 2 shows that L_1 and L_∞ are a bit safer when the attacker knows all the terms of problem (P11), which, in practice, is equivalent to (and as unlikely as) that the attacker knows the original data. Therefore, it can be concluded that, in practice, the three distances have the same low disclosure risk. About the performance in the solution of the optimization problems, L_2 is a priori the most efficient choice, followed by L_1, and finally L_∞. That results from the dimension of the optimization problems to be solved, reported in Table 1. About the quality of the solution obtained, L_1 will provide the best average relative percentage deviation between the original and perturbed table; L_2 the best distance (2-norm); and L_∞ the best maximum relative percentage deviation. This is an immediate consequence of the objective functions considered. In the example at the end of Section 3, L_1 and L_2 were quite similar, and outperformed L_∞. The computational results of next section show that this behaviour is also observed for large instances.

5. Computational results

We implemented and solved the three models described in Sections 2.1–2.3 using the AMPL modelling language (Fourer et al., 1993) and CPLEX 8.0 (ILOG CPLEX, 2002). We applied them to the CSPLIB test suite, the unique currently available set of instances for tabular data protection (Fischetti and Salazar, 2001). CSPLIB can be freely obtained from http://webpages.ull.es/users/casc/#CSPlib:. Although these instances were originally produced for the cell suppression problem, the information provided is the same that for the minimum-distance approach. CSPLIB contains both low-dimensional artificially generated problems, and real-world highly-structured ones. Some of the complex instances were contributed by National Statistical Agencies—as, e.g., Centraal Bureau voor de Statistiek (Netherlands), Energy Information Administration of the Department of Energy (US), Office for National Statistics (United Kingdom) and Statistisches Bumdesant (Germany)—, and therefore are good representatives of theirs real needs. In all the executions a value of at least $a_i + upl_i$ for all $i \in \mathscr{P}$ was imposed (i.e., sense "upper protection" was considered for the sensitive cells), and cell values were weighted by $w_i = 1/a_i$ in the objective function. All runs were carried on a notebook with a Pentium Mobile 4 at 1.8 GHz and 512 Mb of RAM. The problems solved in this section can be obtained in MPS format from http://www-eio.upc.es/~jcastro/data.html. They are delivered to the optimization community as an additional test for linear and quadratic programming solvers.

Table 2 shows the features of the instances considered. The small CSPLIB instances were omitted. Column "Name" shows the instance identifier. Columns "n" and "$|\mathscr{P}|$" provide, respectively, the number of total cells and number of sen-

Table 2
Dimensions of the largest CSPLIB instances

Name	n	$\|\mathcal{P}\|$	m	N.coef
bts4	36,570	2260	36,310	136,912
cbs	11,163	2467	244	22,326
dale	16,514	4923	405	33,028
hier13	2020	112	3313	11,929
hier13 × 13 × 13a	2197	108	3549	11,661
hier13 × 13 × 13b	2197	108	3549	11,661
hier13 × 13 × 13c	2197	108	3549	11,661
hier13 × 13 × 13d	2197	108	3549	11,661
hier13 × 13 × 13e	2197	112	3549	11,661
hier13 × 13 × 7d	1183	75	1443	5369
hier13 × 7 × 7d	637	50	525	2401
hier16	3564	224	5484	19,996
hier16 × 16 × 16a	4096	224	5376	21,504
hier16 × 16 × 16b	4096	224	5376	21,504
hier16 × 16 × 16c	4096	224	5376	21,504
hier16 × 16 × 16d	4096	224	5376	21,504
hier16 × 16 × 16e	4096	224	5376	21,504
jjtabeltest3	3025	1054	1650	7590
nine12	10,399	1178	11,362	52,624
nine5d	10,733	1661	17,295	58,135
ninenew	6546	858	7340	32,920
osorio	10,201	7	202	20,402
table1	1584	146	510	4752
table3	4992	517	2464	19,968
table4	4992	517	2464	19,968
table5	4992	517	2464	19,968
table6	1584	146	510	4752
table7	624	17	230	1872
table8	1271	3	72	2542
targus	162	13	63	360
toy3dsarah	2890	376	1649	9690
two5in6	5681	720	9629	34,310

sitive cells. Column "m" shows the number of constraints. Column "N.coef" gives the number of coefficients of the constraints matrix M. Table 3 shows the results obtained with L_1, L_2 and L_∞. For each distance, the execution time (columns "CPU"), average percentage deviation for all the cells (columns "%Dev."), and two-norm of the deviations vector (columns "2-norm") are provided. The results reported for L_∞ were computed by the simplex method: the interior-point solutions, although with the same objective function, provided worse average percentage deviations and distances for all the instances. The results for L_1 with the simplex and interior-point method were similar, although the simplex was the most efficient choice in most cases—but for the seven most complex instances which are discussed below. The results reported in the Table for L_1 correspond to the simplex solutions, but for the four instances which are clearly marked. In three of these four cases, the simplex method provided a wrong solution. Tuning CPLEX 8.0 we were able to solve them. The interior-point method could solve all the instances with the default settings.

From Table 3 we can conclude that L_1 provides the best mean percentage deviations, since its objective function is exactly the sum of percentage absolute deviations. The L_2 objective provides similar mean percentage deviations but with the lowest two-norms of the deviations vector. This is a consequence of L_2 being the only quadratic objective of the three tested. L_∞ is the slowest option, and does not improve neither the mean percentage deviations nor the 2-norms of the deviations of L_1 or L_2.

The seven more complex instances of CSPLIB are bts4, hier13, hier16, nine12, nine5d, ninenew, and two5in6. Those instances are challenging for other approaches, as cell suppression, whereas, as shown in Table 4, they can be solved in few seconds with the minimum-distance approach. That table reports, for these seven complex instances and each distance, the CPU time required by both the simplex (columns "Simplex") and interior-point algorithms (columns "Int. Point"). We see that, for L_1 and L_∞, the fastest solution algorithm depends on the particular instance, and it is difficult to know in advance which will be the best choice. It is also clear that L_∞ provides the slowest executions, due to the number of extra constraints considered in problem (P9). The L_2 objective, solved through a quadratic interior-point solver, was always the most efficient choice (except for the smallest instance hier13). In most instances the solution time of the L_2 objective was about half the time of the second fastest option. This is because, first, the complexity of solving a quadratic separable optimization problem (i.e., with a diagonal weight matrix W) is the same as that for a linear one, if we use an interior-point algorithm; and second, problem (P5) involves twice as many variables as problem (P6). It is also worth to note that the solution times obtained with the interior-point algorithm, for the three objectives,

Table 3
Results for the largest CSPLIB instances

Name	L_1			L_2			L_∞		
	CPU	$\overline{\%\mathrm{Dev.}}$	2-norm	CPU	$\overline{\%\mathrm{Dev.}}$	2-norm	CPU	$\overline{\%\mathrm{Dev.}}$	2-norm
bts4	16.5	0.7	18,243	11.5	0.8	7912	1594.7	1.1	10,997
cbs	0.0	40.6	75,986	0.1	42.9	55,732	0.1	40.6	75,986
dale	0.7	18.7	4991	0.3	20.3	1859	1.5	21.1	3086
hier13	3.3	0.8	2609	3.8	0.9	2149	5.9	1.0	3504
hier13 × 13 × 13a	1.9	0.8	3094	2.4	0.9	2162	5.9	1.0	3201
hier13 × 13 × 13b	2.0	0.8	3094	2.3	0.9	2162	5.7	1.0	3201
hier13 × 13 × 13c	1.9	0.8	3094	2.5	0.9	2162	5.7	1.0	3201
hier13 × 13 × 13d	2.5	1.6	6187	2.4	1.7	4323	2.5	2.1	7182
hier13 × 13 × 13e	2.5	1.6	6187	2.4	1.7	4323	2.6	2.1	6493
hier13 × 13 × 7d	0.2	0.8	2431	0.3	0.9	1463	0.5	1.1	2588
hier13 × 7 × 7d	0.0	0.9	1850	0.1	1.0	1075	0.1	1.1	2143
hier16	19.9	0.8	3203	17.1	0.9	2706	66.5	1.1	3098
hier16 × 16 × 16a	4.6	0.8	4868	12.0	0.9	2796	33.1	1.0	6053
hier16 × 16 × 16b	4.7	0.8	4868	12.1	0.9	2796	32.9	1.0	6053
hier16 × 16 × 16c	4.7	0.8	4868	12.0	0.9	2796	33.1	1.0	6053
hier16 × 16 × 16d	5.3	1.6	9737	12.0	1.8	5593	46.7	2.2	9337
hier16 × 16 × 16e	5.3	1.6	9737	12.0	1.8	5593	46.9	2.2	9337
jjtabeltest3	0.2	22.1	3.4e + 7	0.1	27.8	2.0e + 7	0.2	30.0	2.7e + 7
nine12	382.1	1.4	5840	18.3	1.5	4878	727.3	1.7	4988
nine5d	126.7	1.7	8316	20.4	1.9	5468	784.5	2.2	5343
ninenew	27.0	1.6	5448	11.1	1.8	4444	199.4	2.2	4708
osorio	0.1	0.03	4	0.2	0.1	3	15.8	0.06	3
table1	0.2[a]	0.9	5.2e + 6	0.0	1.1	2.5e + 6	0.1	1.1	5.3e + 6
table3	0.9	3.0	162,763	0.7	3.5	72,291	12.7	3.8	111,104
table4	0.9	3.0	162,763	0.7	3.5	72,291	12.6	3.8	111,104
table5	1.0	3.0	162,763	0.7	3.5	72,291	12.6	3.8	111,104
table6	0.3[a]	0.9	4.1e + 6	0.0	1.1	2.5e + 7	0.1	1.1	5.3e + 6
table7	0.0	5.9	50,738	0.0	7.2	32,984	0.0	7.5	50,122
table8	0.0	0.0	26	0.0	0.1	15	0.1	0.1	19
targus	0.0[b]	4.1	6958	0.0	4.1	4964	0.0	4.1	6961
toy3dsarah	0.1[a]	2.7	2.4e + 10	0.1	3.0	2.3e + 10	0.0	2.8	2.4e + 10
two5in6	13.6	1.5	4917	9	1.7	3749	83.5	2.0	4137

[a] Simplex provided a wrong solution; interior-point one used.

[b] Best results obtained with the interior-point algorithm.

Table 4
CPU time for the seven most complex instances using the simplex and interior-point algorithms

Instance	L_1		L_2	L_∞	
	Simplex	Int. Point	Int. Point	Simplex	Int. Point
bts4	16.5	39.7	11.5	1594.7	207.0
hier13	3.2	6.9	3.8	5.9	35.2
hier16	19.9	28.4	17.2	66.5	136.9
nine12	382.1	47.4	18.3	727.3	338.8
nine5d	126.7	43.0	20.4	784.5	137.3
ninenew	27.1	24.0	11.2	199.4	120.5
two5in6	13.6	16.9	9.0	83.5	86.5

can even be improved using specialized solvers that exploit the tables structure. Some work has already been done along these lines for L_2 and very large (e.g., up to one million of cells) three-dimensional tables, where the specialized interior-point algorithm of Castro (2000) was two orders of magnitude faster than CPLEX 8.0 (Castro, 2004b).

6. Conclusions

From the computational experiments of this work, the CTA or minimum-distance controlled perturbation framework proved to be an efficient and promising tool for tabular data protection. It was also proved that this class of methods has a low disclosure risk. The three distances tested provided different patterns of deviations, each of them with a clear behaviour. National Statistical Agencies would choose the best suited distance—or some combination of them—for their data. Among the future work to be done we find the development of a heuristic post-process for adjusting, in frequency tables, the possible fractional solutions; and the design of highly efficient interior-point implementations, which should exploit the problem structure. For three-dimensional tables some work has already been done (Castro, 2004b), but it should be extended to general tables.

Acknowledgments

This work was partially supported by the European Union IST-2000-25069 CASC project, and by the Spanish MCyT Project TIC2003-00997. The author thanks Lawrence H. Cox, Ramesh A. Dandekar and the anonymous referees for helpful comments and suggestions.

References

Bacharach, M., 1966. Matrix rounding problems. Management Science 9, 732–742.

Bixby, R.E., 2002. Solving real-world linear programs: A decade and more of progress. Operations Research 50, 3–15.

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.

Castro, J., 2000. A specialized interior-point algorithm for multicommodity network flows. SIAM Journal on Optimization 10 (3), 852–877.

Castro, J., 2002a. Internal communication to partners of the European Union IST-2000-25069 CASC project.

Castro, J., 2002b. Network flows heuristics for complementary cell suppression: an empirical evaluation and extensions. In: Domingo-Ferrer, J. (Ed.), Inference Control in Statistical Databases, Lecture Notes in Computer Science, vol. 2316. Springer, Berlin, pp. 59–73.

Castro, J., 2004a. A fast network flows heuristic for cell suppression in positive tables. In: Domingo-Ferrer, J., Torra, V. (Eds.), Privacy in Statistical Databases, Lecture Notes in Computer Science, vol. 3050. Springer, Berlin, pp. 136–148.

Castro, J., 2004b. Quadratic interior-point methods in statistical disclosure control. Computational Management Science, in press.

Castro, J., 2004c. Computational experiments with minimum-distance controlled perturbation methods. In: Domingo-Ferrer, J., Torra, V. (Eds.), Privacy in Statistical Databases, Lecture Notes in Computer Science, vol. 3050. Springer, Berlin, pp. 73–86.

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

Cox, L.H., 2003. Personal communication.

Cox, L.H., Ernst, L.R., 1982. Controlled rounding. INFOR 20, 423–432.

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

Cox, L.H., Kelly, J.P., Patil, R., 2004. Balancing quality and confidentiality for multivariate tabular data. In: Domingo-Ferrer, J., Torra, V. (Eds.), Privacy in Statistical Databases, Lecture Notes in Computer Science, vol. 3050. Springer, Berlin, pp. 87–98.

Dandekar, R.A., 2003a. Cost effective implementation of synthetic tabulation (a.k.a. controlled tabular adjustments) in legacy and new statistical data publication systems. Presented at the Joint ECE/Eurostat Work Session on Statistical Data Confidentiality, Luxembourg. Available from: <http://www.unece.org/stats/documents/2003.04.confidentiality. htm>.

Dandekar, R.A., 2003b. Personal communication.

Dandekar, R.A., Cox, L.H., 2002. Synthetic tabular data: An alternative to complementary cell suppression, manuscript, Energy Information Administration, US Department of Energy. Available from the first author on request (Ramesh.Dandekar@eia.doe.gov).

Dantzig, G.B., 1963. Linear Programming and Extensions. Princeton University Press, Princeton.

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

Domingo-Ferrer, J. (Ed.), 2002. Inference Control in Statistical Databases, Lecture Notes in Computer Science, vol. 2316. Springer, Berlin.

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 (5), 545–556.

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

Fischetti, M., Salazar, J.J., 2001. Solving the cell suppression problem on tabular data with linear constraints. Management Science 47 (7), 1008–1026.

Fourer, R., Gay, D.M., Kernighan, B.W., 1993. AMPL: A Modeling Language for Mathematical Programming. Boyd & Fraser, Danvers, MA.

ILOG CPLEX, 2002. ILOG CPLEX 8.0 Reference Manual Library, Gentilly, France: ILOG.

Kelly, J.P., Assad, A.A., Golden, B.L., 1990a. The controlled rounding problem: Relaxations and complexity issues. OR Spektrum 12, 129–138.

Kelly, J.P., Golden, B.L., Assad, A.A., 1990b. Using simulated annealing to solve controlled rounding problems. Annals of Operations Research 2 (2), 174–190.

Kelly, J.P., Golden, B.L., Assad, A.A., Baker, E.K., 1990c. Controlled rounding of tabular data. Operations Research 38 (5), 760–772.

Kelly, J.P., Golden, B.L, Assad, A.A., 1992. Cell suppression: Disclosure protection for sensitive tabular data. Networks 22, 28–55.

Luenberger, D.G., 1989. Linear and Nonlinear Programming, second ed. Addison Wesley, Reading, MA.

Robertson, D.A., Ethier, R., 2002. Cell suppression: Experience and theory. In: Domingo-Ferrer, J. (Ed.), Inference Control in Statistical Databases, Lecture Notes in Computer Science, vol. 2316. Springer, Berlin, pp. 8–20.

Willenborg, L., de Waal, T., (Eds.), 2000. Elements of Statistical Disclosure Control, Lecture Notes in Statistics, vol. 155. Springer, New York.

Wright, S.J., 1997. Primal–Dual Interior-Point Methods. SIAM, Philadelphia.