Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Computers & Operations Research 32 (2005) 2919–2936

computers & operations research

www.elsevier.com/locate/dsw

Exact disclosure prevention in two-dimensional statistical tables

M.T. Almeida[a,b], F.D. Carvalho[a,b,*]

[a] Instituto Superior de Economia e Gestão—Universidade Tècnica de Lisboa, Lisboa, Portugal
[b] Centro de Investigação Operacional-FCUL, Portugal

Avaliable online 24 June 2004

Abstract

We propose new formulations for the exact disclosure problem and develop Lagrangian schemes, that rely on shortest path problems, to generate near optimal solutions. Computational experience is reported for 550 tables with up to 40,000 cells. A proven optimal solution was obtained for 95% of the instances and a near optimal solution was computed for each remaining instance as well as an upper bound on the deviation from the optimum.

Keywords: Statistical disclosure; Cell suppression; Graph theory; Integer programming; Lagrangian relaxation

1. Introduction

Statistical offices disseminate economic and social data using two-dimensional tables built from microdata aggregating figures by relevant categories (economic activity, geographic region, company number of employees, etc.). The row and column subtotals are presented in the last column and the last row, respectively. This information is redundant when all table values are published but may be important otherwise. Many statistical tables display frequency counts and therefore do not contain negative values. These tables are called nonnegative [1] (some authors call them positive [2,3]). Economic tables often contain magnitude data that may be unrestricted in sign (e.g. financial deficits or negative growth rates). Tables containing data unrestricted in sign are called general tables [1,2,4].

* Corresponding author. Departamento de Matemática, Instituto Superior de Economia e Gestão, Rua do Quelhas, 1200-781 Lisboa, Portugal. Fax: 00-351-213922781.
E-mail address: filipadc@iseg.utl.pt (F.D. Carvalho).

0305-0548/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2004.04.015

Statistical offices must guarantee that the tables made available to the public do not enable an intruder to identify a respondent through its responses.

The first issue is then to determine the cells that reveal too much information about individual respondents and shall be considered sensitive. The disclosure risk of a cell depends on its contents, on whether it results from data gathered from the whole population or from some population sample, on whether the disclosure control rules are known to the intruders or not, etc. For a good discussion on risk disclosure for tabular data see chapter 6 of [1]. In this work we will assume that the set of sensitive cells is given.

To protect a sensitive cell against disclosure one may suppress it from publication or make some sort of perturbation to its value (by adding noise, by stochastic rounding, etc.) [1,5]. For tables that aggregate magnitude data, such as economic census tables, suppression is the preferred technique [2]. Due to the table additive structure, suppressing from publication only the sensitive values does not, in general, ensure their protection. It is then necessary to omit also some nonsensitive values, known as complementary suppressions, which yields an undesirable loss of information to the user community.

So, in very general terms, the problem is that of finding a set of complementary suppressions that protects sensitive entries avoiding unnecessary losses of nonsensitive information.

Exact disclosure is typically used for frequency count data [2], when sensitive values are considered protected if and only if their exact values cannot be deduced from the published tables. Even if their exact value is impossible to compute, combining the additive structure with the nonnegativity conditions may allow an intruder to compute narrow ranges for sensitive figures. If these ranges are deemed unacceptable a different protection criterion is adopted: a protection interval is defined a priori, for each sensitive cell, and the table considered safe for public release if and only if the ranges of values that can be deduced for the sensitive values, by any intruder, contain the corresponding protection intervals. The protection intervals may depend on each cell value [6–8] or may be the same for all sensitive cells [3]. In [6] a sliding version of protection intervals is proposed, fixing just the width of the protections. For integer nonnegative data if all widths are set to one this amounts to exact disclosure protection. For general tables, the magnitude of the sensitive values is irrelevant for their protection. Either it is possible to compute the exact value of a suppressed cell or no meaningful estimation is possible for it, regardless of its value. In other words, for general tables only exact disclosure must be prevented.

In the open literature the case of nonnegative tables with protection intervals has received more attention than the exact disclosure criterion. Heuristic methods and lower bounding procedures have been developed by several authors [2,3,6,8–10]. Recently, Fischetti and Salazar [7] proposed a branch-and-cut approach based on a new network flow formulation, with an exponential number of cut type constraints, to which constraints selected from five new families of valid inequalities are added. As far as we are aware, this is the best exact algorithm to date for nonnegative tables with the fixed interval protection criterion. The method may also be used for general tables but does not solve the problem with sliding protection ranges.

For the exact disclosure problem Gusfield [11] devised a polynomial time algorithm that solves to optimality the special case of minimizing the number of complementary suppressions in strictly positive integer tables, under the condition that the intruders are not aware of the fact that the tables do not contain zero cells. The core of his method is the solution of a graph augmentation problem on a mixed graph representation of statistical tables, assuming that no subtotal cell may be suppressed. The method does not solve the more general case addressed in this paper.

In this paper, we address the exact disclosure problem in general and in integer nonnegative tables with arbitrary nonnegative weights associated with the nonsensitive cells (to quantify the loss of information due to their suppression) and no special conditions on the subtotals. The problem is *NP*-hard as its restricted version with all weights equal to one is known to be *NP*-hard [11]. The main contributions of this work are two new compact linear integer formulations, one for general tables and one for nonnegative integer tables, and a Lagrangian scheme to decompose them into shortest path problems. As the formulations are compact (i.e., have a polynomial number of variables and of constraints) they can be solved by commercial packages for many small and medium sized tables within reasonable computing times. The Lagrangian scheme produces lower bounds on the optimum values that are theoretically equal to the linear bounds, for general tables, and that dominate the linear bounds for nonnegative tables. Coupled with a constructive heuristic it also generates near optimal solutions and upper bounds on the gaps between the best heuristic value and the optimum. As the Lagrangian procedure relies on the solution of shortest path subproblems it is very easy to implement and rather fast. The computational experience showed that the Lagrangian scheme often generates proven optimal solutions. This approach is theoretically compact, quite efficient and flexible to implement. These are desirable features for methods to be adopted in practical settings, involving personnel with different technical backgrounds [2].

The paper is organized as follows. In Section 2 we set the notation used in the sequel and review a graph representation of the problem. In Sections 3 and 4 we present the new approach for general and for nonnegative tables. Section 5 reports the computational results and Section 6 contains the final remarks.

2. The exact disclosure problem

A two-dimensional statistical table $A=[a_{ij}]$ may be defined as an $(m+1)\times(n+1)$ array of real numbers. The values in the $(m+1)$th row are the column subtotals and the values in the $(n+1)$th column are the row subtotals:

$$a_{m+1,j}=\sum_{i=1}^{m} a_{ij},\quad j=1,\dots,n,$$

$$a_{i,n+1}=\sum_{j=1}^{n} a_{ij},\quad i=1,\dots,m.$$

The value $a_{m+1,n+1}$ is the grand total:

$$a_{m+1,n+1}=\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}.$$

If all entries in A are nonnegative, A is said to be a nonnegative table. Otherwise A is called a general table. In any case we will assume that there is at least one nonzero cell in A.

Given a statistical table A let

$$\mathscr{C}=\{(i,j):1\leqslant i\leqslant m+1,\ 1\leqslant j\leqslant n+1\}$$

be the set of its cells and let

$$S_1 = \{(i_k, j_k) : 1 \leqslant k \leqslant p\}$$

be the set of sensitive cells (also known as primary suppressions or confidential cells). Sets of complementary suppressions will be denoted by S_2.

A set S_2 is said to be feasible if omitting the values in $S_1 \cup S_2$ precludes an intruder from computing the values in S_1.

The total loss of information associated with a set S_2 of complementary suppressions is given by

$$\xi(S_2) = \sum_{(i,j)\in S_2} \xi_{ij},$$

where ξ_{ij} are weights associated with cells in $\mathscr{C} \setminus S_1$ to quantify the loss of information due to the suppression of their values from publication. If all weights are set to 1, then minimizing $\xi(S_2)$ yields the minimum cardinality feasible set S_2. However omitting large values is, in general, considered more undesirable than omitting small values. So several authors set $\xi_{ij} = |a_{ij}|$ for every cell in $\mathscr{C} \setminus S_1$.

A statistical table may be represented by an undirected bipartite graph $G = (V, \mathscr{E})$ [4]. The node set, $V = R \cup C$, is the union of a set R of $m+1$ nodes, representing the table rows, with a set C of $n+1$ nodes, representing the table columns. Each edge $(i, j) \in \mathscr{E}$ represents a table cell. To represent only the suppressed cells one may use the subgraph $G_{S_1 \cup S_2} = (V, S_1 \cup S_2)$. From now on we will make no distinction between an edge (i, j) and the cell it represents. In the sequel we will assume the reader is familiar with graph and network terminology and basic results. For a good reference see [12].

3. General tables

For general tables the protection against disclosure may be established based on the definition of a table completion given in [4]. A completion of a table $A = [a_{ij}]$, $(i, j) \in \mathscr{C}$, is a set of values $D = [d_{ij}]$, $(i, j) \in \mathscr{C}$, for cells in $\mathscr{C}$, which is consistent with the given row and column subtotals, and such that $d_{ij} = a_{ij}$ for all unsuppressed cells $(i, j) \in \mathscr{C} \setminus (S_1 \cup S_2)$. Exact disclosure occurs in a table A if for some sensitive cell $(i_k, j_k) \in S_1$, there are no two completions D and D' such that $d_{i_k j_k} \neq d'_{i_k j_k}$.

In a general table with a set of sensitive cells S_1 and a set of complementary suppressions S_2, the value of a sensitive cell $(i, j) \in S_1$ is protected against exact disclosure if and only if its corresponding edge belongs to a circuit in the corresponding subgraph $G_{S_1 \cup S_2} = (V, S_1 \cup S_2)$ [4]. Since general tables are not restricted in sign for them exact disclosure protection implies that no meaningful estimation for suppressed values can be derived from the published values.

3.1. Integer formulation

Let A be a general table with a set $S_1 = \{(i_k, j_k) : 1 \leqslant k \leqslant p\}$ of p sensitive cells. If a path of suppressed cells is created from j_k to i_k, for each cell $(i_k, j_k) \in S_1$, then each sensitive cell is embedded in a circuit of suppressions and the table is safe for release. So the problem for general tables may be restated as follows: find a minimum cost set of p paths in the undirected bipartite graph $G = (R \cup C, \mathscr{E})$, each linking a node j_k to a node i_k, and excluding edge (i_k, j_k), $1 \leqslant k \leqslant p$.

This problem may be formulated as a linear programming problem with binary variables. For reasons to be explained later, we will look for directed paths. So a pair of reverse arcs, (i, j) and (j, i), will be associated to each edge (i, j) in $\mathscr{E}$.

Consider the following variables:

$$\underset{(i,j)\in\mathscr{C}\setminus S_1}{x_{ij}} = \begin{cases} 1 & \text{if } (i,j)\in S_2, \\ 0 & \text{otherwise}, \end{cases}$$

$$\underset{\substack{i\neq i_k, j\neq j_k, k=1,\ldots,p\\ i\in R, j\in C}}{y^k_{ij}} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is in a directed path from } j_k \text{ to } i_k, \\ 0 & \text{otherwise}, \end{cases}$$

$$\underset{\substack{(j,i)\neq(j_k,i_k), k=1,\ldots,p\\ j\in C, i\in R}}{y^k_{ji}} = \begin{cases} 1 & \text{if arc } (j,i) \text{ is in a directed path from } j_k \text{ to } i_k, \\ 0 & \text{otherwise}. \end{cases}$$

A minimum cost feasible set of complementary suppressions is an optimal solution for the following binary problem:

$$(\text{P}) \quad Z_{\text{P}} = \min \sum_{(i,j)\in\mathscr{C}\setminus S_1} \xi_{ij} x_{ij}, \tag{1}$$

s.t.

$$x_{ij}\in\{0,1\},\ (i,j)\in\mathscr{C}\setminus S_1$$

$$\text{for } k = 1,\ldots,p, \tag{2}$$

$$\sum_{i\in R} y^k_{j_k i} = 1, \tag{3}$$

$$\sum_{j\in C} y^k_{j i_k} = 1, \tag{4}$$

$$\sum_{j\in C} y^k_{ji} - \sum_{j\in C} y^k_{ij} = 0, \quad i\in[R\setminus\{i_k\}], \tag{5}$$

$$\sum_{i\in R} y^k_{ij} - \sum_{j\in R} y^k_{ji} = 0, \quad j\in[C\setminus\{j_k\}], \tag{6}$$

$$y^k_{ij} \leqslant x_{ij}, \quad (i,j)\in\mathscr{C}\setminus S_1, \tag{7}$$

$$y^k_{ji} \leqslant x_{ij}, \quad (i,j)\in\mathscr{C}\setminus S_1, \tag{8}$$

$$y^k_{ij}, y^k_{ji} \in\{0,1\}, \quad (i,j)\in\mathscr{C}. \tag{9}$$

Constraints (3)–(6), (9) define, for every sensitive cell (i_k, j_k), $k\in\{1,\ldots,p\}$, a directed path from node j_k to node i_k that does not contain arc (j_k, i_k). The coupling constraints (7) and (8) along with (2) ensure that a cell is suppressed whenever one of its corresponding arcs is used in a path.

The coupling constraints may be substituted by

$$y_{ij}^k + y_{ji}^k \leqslant x_{ij} \quad (i,j) \in \mathscr{C} \setminus S_1. \tag{10}$$

As in some optimal solution at most one arc in the pair of reverse arcs is in every path this substitution does not alter the value of Z_{P}. With (10) the number of coupling constraints is reduced by 50% and so is the number of variables in the Lagrangian dual problem defined below.

The numbers of variables and of constraints in (P) are polynomial in the table's size. So, for many small and medium sized tables (P) may be solved to optimality, within reasonable computing time, by any integer programming commercial package. When it is not the case, the value of its linear programming relaxation is easily obtained if a Lagrangian relaxation approach is adopted. It is also easy to generate feasible near optimal solutions for (P) based on Lagrangian solutions adapting the procedure proposed by Carvalho et al. [4].

3.2. Lagrangian approach

Relaxing the coupling constraints (10) in a Lagrangian fashion [13] yields the following Lagrangian relaxation problem, for each vector of Lagrangian multipliers, $\boldsymbol{\lambda} \geqslant \mathbf{0}$:

$$(\mathrm{P}_{\lambda}) \quad z(\boldsymbol{\lambda}) = \min \sum_{(i,j)\in\mathscr{C}\setminus S_1} \xi_{ij} x_{ij} + \sum_{k=1}^{p} \sum_{(i,j)\in\mathscr{C}\setminus S_1} \lambda_{ij}^k \, (y_{ij}^k + y_{ji}^k - x_{ij})$$

$$\text{s.t. } (2),\ (3),\ (4),\ (5),\ (6),\ (9).$$

The best vector of multipliers is an optimal solution of the Lagrangian dual:

$$(\mathrm{D}) \quad \operatorname*{Max}_{\boldsymbol{\lambda} \geqslant \mathbf{0}} z(\boldsymbol{\lambda}).$$

As (P_{λ}) has the integrality property the optimum value of (D) is equal to the optimum value of the linear relaxation of (P) [13]. As shown in the appendix directing the path variables strengthens the linear bound even when compared to an undirected formulation to which some cuts are added (for a more detailed presentation see [14]). This is the reason to associate a pair of directed arcs (i,j) and (j,i), to each edge (i,j) in $\mathscr{E}$ and then look for directed paths. Problem (P_{λ}) is very fast to solve because it can be decomposed into $(p+1)$ independent subproblems: one on the x_{ij} variables only, that is solved by inspection, and p shortest path problems with nonnegative arc lengths, for which very efficient algorithms are well known [12].

4. Nonnegative integer tables

If it is known that all table values are nonnegative, including a sensitive cell in a circuit of suppressions may not be sufficient to prevent the disclosure of its exact value. Let $(1,1)$ be the only sensitive cell in the table of Fig. 1.

Suppressing also cells in $S_2 = \{(1,3),(3,1),(3,3)\}$ includes cell (1,1) in a circuit as shown in Fig. 2.

However, due to the nonnegativity constraints, from column 3 an intruder deduces that $a_{13} = a_{33} = 0$ and then, from row 1, deduces that the sensitive value, a_{11}, must be 2. In other words, in all

	c_1	c_2	c_3	c_4
r_1	2	3	0	5
r_2	1	4	1	6
r_3	0	5	0	5
r_4	3	12	1	16

Fig. 1.

	c_1	c_2	c_3	c_4
r_1	*	3	*	5
r_2	1	4	1	6
r_3	*	5	*	5
r_4	3	12	1	16

Fig. 2.

	c_1	c_2	c_3	c_4
r_1	2	3	0	5
r_2	1	4	1	6
r_3	0	5	1	6
r_4	3	12	2	17

Fig. 3.

	c_1	c_2	c_3	c_4
r_1	2	3	1	6
r_2	1	4	1	6
r_3	1	5	0	6
r_4	4	12	2	18

Fig. 4.

completions, $D=[d_{ij}]$, $(i,j)\in\mathscr{C}$, the condition $d_{11}=2$ must hold to comply with $d_{ij}\geqslant 0$ for all $(i,j)\in\mathscr{C}$.

Consider now tables in Figs. 3 and 4, again with only one sensitive cell, (1,1).

If the same set of complementary suppressions is selected then both tables get protected against exact disclosure of the value in cell (1,1). To confirm that, consider the following completions D and D', given in Figs. 5 and 6, respectively, with $d_{11}\neq 2$ and $d'_{11}\neq 2$.

	c_1	c_2	c_3	c_4
r_1	1	3	1	5
r_2	1	4	1	6
r_3	1	5	0	6
r_4	3	12	2	17

Fig. 5.

	c_1	c_2	c_3	c_4
r_1	3	3	0	6
r_2	1	4	1	6
r_3	0	5	1	6
r_4	4	12	2	18

Fig. 6.

Note that for this set S_2 of complementary suppressions there is no completion for the table in Fig. 3 with $d_{11} > 2$ and there is no completion for the table in Fig. 4 with $d'_{11} < 2$.

The examples above show that, for nonnegative tables, the same suppression pattern may or may not render a table safe for release, depending on the cell values.

In general terms, in a nonnegative integer table A, with a set $S_1 \cup S_2$ of suppressions, the value of a positive sensitive cell (i_k, j_k) cannot be exactly disclosed if and only if there is either a completion D for which $d_{i_k j_k} \leqslant a_{i_k j_k} - 1$ or a completion D' for which $d'_{i_k j_k} \geqslant a_{i_k j_k} + 1$. If $a_{i_k j_k} = 0$ then the sensitive cell (i_k, j_k) cannot be exactly disclosed if and only if there is a completion D for which $d_{i_k j_k} \geqslant a_{i_k j_k} + 1$, as values $d_{i_k j_k} \leqslant a_{i_k j_k} - 1$ would be negative.

So, in this case, the approach proposed in the previous section is no longer valid. To deal with the nonnegative case a directed graph, $H = (R \cup C, F)$, will be built excluding from its arc set, F, a subset of arcs, E, that would otherwise be associated with zero cells. Note that in a completion D if, for an internal sensitive cell (i_k, j_k), $d_{i_k j_k} > a_{i_k j_k}$ then either $(i_k, n+1)$ and $(m+1, j_k)$ are both suppressed or, to keep the additive structure in row i_k and column j_k, there must be some internal cells for which $d_{ij} < a_{ij}$. But, due to the nonnegativity conditions, $d_{ij} < a_{ij}$ is feasible only if $a_{ij} > 0$. A similar argument applies to the subtotal cells. As a consequence the set of excluded arcs is

$$\begin{aligned} E = &\{(j,i) : 1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n,\ a_{ij} = 0\} \\ &\cup \{(i, n+1) : 1 \leqslant i \leqslant m,\ a_{i,n+1} = 0\} \\ &\cup \{(m+1, j) : 1 \leqslant j \leqslant n,\ a_{m+1,j} = 0\} \end{aligned}$$

and the arc set in H is $F = [(R \times C) \cup (C \times R)] \setminus E$.

A set of complementary suppressions, S_2, is feasible for a nonnegative integer table A, with a set S_1 of sensitive cells, if and only if for each positive cell (i_k, j_k) in S_1 there is either a path in the subgraph $H_{S_1 \cup S_2}$ of H from j_k to i_k that does not include arcs (i_k, j_k) and (j_k, i_k) or a path from i_k to j_k that does not include arcs (i_k, j_k) and (j_k, i_k). In the same subgraph for each zero cell (i_k, j_k) in S_1 there must be a path from j_k to i_k that does not include arcs (i_k, j_k) and (j_k, i_k).

In the examples above with $S_2 = \{(1,3),(3,1),(3,3)\}$, for the table in Fig. 3 there is a path in $H_{S_1 \cup S_2}$ from row node 1 to column node 1 through (3,3) and no path from column node 1 to row node 1; on the contrary, for the table in Fig. 4 there is a path from column node 1 to row node 1 through (3,3) and no path from row node 1 to column node 1. Note that the either/or condition is not exclusive: for example, with the suppression set $\{(1,2),(2,1),(1,2)\}$ there are in $H_{S_1 \cup S_2}$ paths from column node 1 to row node 1 and from row node 1 to column node 1 for both tables.

4.1. Integer formulation

The same rationale underlying formulation (P), for general tables, may be used to state a formulation (P_N), for nonnegative integer tables, defining two directed paths for each positive sensitive cell and one directed path to each zero sensitive cell.

Let $K = \{1, \ldots, p\}$ be partitioned into $K_1 = \{k \in K : a_{i_k j_k} = 0\}$ and $K_2 = \{k \in K : a_{i_k j_k} > 0\}$. Consider the binary variables:

x_{ij} $(i,j) \in \mathscr{C} \setminus S_1$ defined as before

$$\alpha^k = \begin{cases} 1 & \text{if a path is generated from } j_k \text{ to } i_k, \\ 0 & \text{otherwise,} \end{cases} \qquad k \in K_2,$$

$$\beta^k = \begin{cases} 1 & \text{if a path is generated from } i_k \text{ to } j_k, \\ 0 & \text{otherwise,} \end{cases} \qquad k \in K_2$$

for $(i,j),(j,i) \in F$:

$$\underset{\substack{i \neq i_k,\ j \neq j_k \\ i \in R, j \in C}}{y_{ij}^k} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is in a directed path from } j_k \text{ to } i_k, \\ 0 & \text{otherwise,} \end{cases} \qquad k \in K,$$

$$\underset{\substack{(j,i) \neq (j_k, i_k) \\ j \in C, i \in R}}{y_{ji}^k} = \begin{cases} 1 & \text{if arc } (j,i) \text{ is in a directed path from } j_k \text{ to } i_k, \\ 0 & \text{otherwise,} \end{cases} \qquad k \in K,$$

$$\underset{\substack{(i,j) \neq (i_k, j_k) \\ i \in R, j \in C}}{w_{ij}^k} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is in a directed path from } i_k \text{ to } j_k, \\ 0 & \text{otherwise,} \end{cases} \qquad k \in K_2,$$

$$\underset{\substack{j \neq j_k,\ i \neq i_k \\ j \in C, i \in R}}{w_{ji}^k} = \begin{cases} 1 & \text{if arc } (j,i) \text{ is in a directed path from } i_k \text{ to } j_k, \\ 0 & \text{otherwise,} \end{cases} \qquad k \in K_2.$$

And the binary problem:

$$(\mathrm{P}_N) \quad Z_{\mathrm{P}_N} = \min \sum_{(i,j) \in \mathscr{C} \setminus S_1} \xi_{ij} x_{ij} \tag{11}$$

s.t.

$$\sum_{i\in R} y^k_{j_k i} = \alpha^k, \quad k\in K_2, \tag{12}$$

$$\sum_{j\in C} y^k_{j i_k} = \alpha^k, \quad k\in K_2, \tag{13}$$

$$\sum_{i\in R} y^k_{j_k i} = 1, \quad k\in K_1, \tag{14}$$

$$\sum_{j\in C} y^k_{j i_k} = 1, \quad k\in K_1, \tag{15}$$

$$\sum_{j\in C} y^k_{ji} - \sum_{j\in C} y^k_{ij} = 0, \quad i\in[R\setminus\{i_k\}],\ k\in K, \tag{16}$$

$$\sum_{i\in R} y^k_{ij} - \sum_{i\in R} y^k_{ji} = 0, \quad j\in[C\setminus\{j_k\}], \quad k\in K, \tag{17}$$

$$y^k_{ij} + y^k_{ji} \leqslant x_{ij}, \quad (i,j)\in\mathscr{C}\setminus S_1,\ k\in K, \tag{18}$$

$$\sum_{j\in C} w^k_{i_k j} = \beta^k, \quad k\in K_2, \tag{19}$$

$$\sum_{i\in R} w^k_{i j_k} = \beta^k, \quad k\in K_2, \tag{20}$$

$$\sum_{j\in C} w^k_{ji} - \sum_{j\in C} w^k_{ij} = 0, \quad i\in[R\setminus\{i_k\}],\ k\in K_2, \tag{21}$$

$$\sum_{i\in R} w^k_{ij} - \sum_{i\in R} w^k_{ji} = 0, \quad j\in[C\setminus\{j_k\}],\ k\in K_2, \tag{22}$$

$$w^k_{ij} + w^k_{ji} \leqslant x_{ij}, \quad (i,j)\in\mathscr{C}\setminus S_1,\ k\in K_2, \tag{23}$$

$$\alpha^k + \beta^k \geqslant 1,\ k\in K_2, \tag{24}$$

$$\beta^k \in \{0,1\}, \quad k\in K_2, \tag{25}$$

$$\alpha^k \in \{0,1\}, \quad k\in K_2, \tag{26}$$

$$x_{ij} \in \{0,1\}, \quad (i,j)\in\mathscr{C}\setminus S_1, \tag{27}$$

$$y^k_{ij}, y^k_{ji} \in \{0,1\}, \quad (i,j)\in\mathscr{C}\setminus S_1,\ k\in K, \tag{28}$$

$$w^k_{ij}, w^k_{ji} \in \{0,1\}, \quad (i,j)\in\mathscr{C}\setminus S_1,\ k\in K_2. \tag{29}$$

As all the weights ξ_{ij} are nonnegative there is an optimal solution to the binary problem where all constraints (24) are tight, i.e., where

$$\alpha^k + \beta^k = 1, \quad k\in K_2.$$

Therefore variables β^k, $k\in K_2$, and constraints (24), (25) may be dropped from the model, with (19) and (20) replaced with

$$\sum_{j\in C} w^k_{i_k j} = 1-\alpha^k, \quad k\in K_2, \tag{30}$$

$$\sum_{i\in R} w^k_{i j_k} = 1-\alpha^k, \quad k\in K_2. \tag{31}$$

To strengthen the linear relaxation, the coupling constraints (18) and (23) may be written in the lifted form:

$$y^k_{ij} + y^k_{ji} \leqslant x_{ij}, \quad (i,j)\in \mathscr{C}\setminus S_1,\ k\in K_1, \tag{32}$$

$$y^k_{ij} + y^k_{ji} + w^k_{ij} + w^k_{ji} \leqslant x_{ij}, \quad (i,j)\in \mathscr{C}\setminus S_1,\ k\in K_2. \tag{33}$$

From now on we will consider the feasible region of P_N defined by constraints (12)–(17), (30), (31), (21), (22), (32), (33), (26)–(29).

4.2. Lagrangian approach

Let $|K_2|=\eta$. To each vector $\boldsymbol{\alpha}=[\alpha^1,\ldots,\alpha^\eta]\in\{0,1\}^\eta$ one can associate a problem $\mathrm{P}_N(\boldsymbol{\alpha})$, setting the values of the α^k variables to their values in $\boldsymbol{\alpha}$. By construction, the optimum value of P_N is

$$Z^*_{\mathrm{P}_N} = \min\{Z^*_{\mathrm{P}_N}(\boldsymbol{\alpha}):\ \boldsymbol{\alpha}\in\{0,1\}^\eta\}.$$

In each problem $\mathrm{P}_N(\boldsymbol{\alpha})$ relaxing constraints (32) and (33) in a Lagrangian fashion, with a vector of multipliers $\boldsymbol{\mu}$, yields a relaxation problem, $\mathrm{P}_N(\boldsymbol{\alpha},\boldsymbol{\mu})$, that may be decomposed into $p+1$ subproblems as follows:

(i) one subproblem defined on the x_{ij} variables by constraints (27);
(ii) $(p-\eta)$ independent shortest path problems (one for each zero sensitive cell) defined on y^k_{ij} and y^k_{ji}, $k\in K_1$, by constraints (14), (15) and by constraints (16), (17) and (28) for $k\in K_1$;
(iii) η independent subproblems (one for each positive sensitive cell) defined on α^k, y^k_{ij}, y^k_{ji}, w^k_{ij}, w^k_{ji}, $k\in K_2$, by constraints (12), (13), (30), (31), (21), (22), (26), (29) and by constraints (16), (17) and (28) for $k\in K_2$.

For each subproblem in (iii), i.e., for each $k\in K_2$, in any feasible solution either $\alpha^k=1$ or $\alpha^k=0$. If $\alpha^k=1$ then constraints (30), (31), (21), (22) and (29) are trivially fulfilled by the null vector and the subproblem reduces to a shortest path problem from j_k to i_k; if $\alpha^k=0$ then constraints (12), (13), (16), (17) and (28) are trivially fulfilled by the null vector and the subproblem reduces to a shortest path problem from i_k to j_k. So the optimum solution of each subproblem in (iii) may be obtained solving two shortest path problems, keeping the variable values for the shortest one and setting all other variables to zero.

The best value for vector of multipliers $\boldsymbol{\mu}$ is obtained solving the Lagrangian dual. The optimum of the Lagrangian dual is equal to the optimum of the mixed integer problem that results from (P_N) substituting constraints (27)–(29) by their linear relaxation [13]. As a consequence the lower bound on the optimum of (P_N) generated by this procedure is tighter than the bound obtained with the linear relaxation of (P_N).

5. Computational experience

The computational study of the methods proposed in Sections 3 and 4 was carried out on a PC PentiumIII with 128 Mbyte RAM. The integer formulations were solved with the commercial package Cplex 8.1.0. with a time limit of 1 h to solve each problem. To implement the Lagrangian approaches we programmed codes in Pascal with Delphi-32 development environment editor. The shortest path problems resulting from the Lagrangian relaxations were computed with Dijkstra's algorithm [12]. The Lagrangian duals were solved by a standard implementation of subgradient optimization [15].

The number of variables and constraints in formulations (P) and (P_N), as well as the number of variables in the Lagrangian duals, depend on the table dimensions and on the number of cells that require protection. Note that all, if any, sensitive cells that belong to at least one circuit in the sensitive cell supporting network do not require complementary suppressions for their protection. These cells are identified and removed from S_1 in a preprocessing phase. The cells remaining in S_1 at the end of the preprocessing phase are called unsafe cells.

A heuristic similar to the one proposed in [4] was embedded in the Lagrangian procedure to generate feasible solutions. It is first called, before starting the subgradient optimization algorithm, with the original weight values, to generate an initial incumbent solution. After that, it is periodically called during the subgradient optimization stage, with weights $\xi_{ij} = 0$ assigned to all variables x_{ij}, $(i, j) \in \mathscr{C} \setminus S_1$, equal to one in the current Lagrangian relaxation solution. The incumbent solution is updated whenever a better solution is found. At the end of the Lagrangian computations, if the incumbent value is equal to the dual optimum, the final incumbent is a proven optimal solution. In this case we call it an in-proven optimal solution. Otherwise the Lagrangian scheme yields an upper bound on the gap between the final incumbent value and the optimum but does not allow any optimality claim by itself. In this case we call the final incumbent an out-proven optimal solution if we are able to find, with the Cplex package, an optimum to the corresponding integer model that matches its value.

To evaluate the computational performance of the methods two data sets were considered.

In the first data set 210 tables with dimensions from 20×10 up to 200×200 were randomly generated reproducing the tables in [7] suggested by Dr. Roberto Benedetti of the Italian Statistical Office. Since only strictly positive suppressions are allowed in this case, it is easy to establish that, with the variables associated with zero valued cells fixed at zero, formulation (P) for general tables is valid for the exact disclosure problem in this class of instances, following the explanation given in Section 4.

In the second data set 170 tables, with dimensions from 50×10 up to 100×100, were generated following the rules used in [6] but decreasing the density of sensitive cells to 25% of the density adopted in [6] to increase the proportion of unsafe cells and generate more challenging problems.

Table 1 shows the information given in Tables 2–4, where each row refers to results obtained over a set of 10 instances.

The computational results obtained with the first data set are shown in Table 2. The effectiveness of the preprocessing phase increased with table dimensions and was especially impressive for square tables. All instances were solved to optimality. With Cplex the optimum for the integer formulation was found in 62.9% of the tables. The Lagrangian procedure produced 98% of in-proven optimal solutions and all final incumbents were proven optimal. The Lagrangian procedure took, on average,

Table 1

$m+1$	Number of rows (including marginal total)
$n+1$	Number of columns (including marginal total)
p	Average number of sensitive cells
up	Average number of unsafe cells
minup	Minimum number of unsafe cells
maxup	Maximum number of unsafe cells
Cpl	Total number of optimal solutions obtained with Cplex
Lag	Total number of in-proven optimal solutions obtained with the Lagrangian procedure
IGap	Average deviation of the initial incumbent value from the Lagrangian bound
Fgap	Average deviation of the final incumbent value from the Lagrangian bound
T(s)	Average wall clock seconds taken by the Lagrangian procedure
Cpl+Lag	Number of in-proven and out-proven optimal solutions

Table 2
Results on the first set of tables

$m+1$	$n+1$	p	up	minup	maxup	Cpl	Lag	IGap	FGap	T (s)	Cpl+Lag
20	10	7.60	7.60	3	14	10	9	7.12	0.39	0.15	10
20	20	12.40	12.40	5	16	10	9	13.84	0.07	2.85	10
40	10	13.00	13.00	10	18	10	10	4.56	0.00	0.92	10
40	20	28.40	27.40	21	33	10	9	3.49	0.06	22.29	10
40	30	46.70	32.90	15	45	10	9	9.32	0.10	64.31	10
40	40	65.80	37.30	29	59	10	10	5.23	0.00	121.54	10
60	20	43.10	36.60	29	47	10	10	3.95	0.00	59.39	10
60	40	90.30	36.90	29	44	10	10	8.67	0.00	137.87	10
60	60	141.60	31.20	22	44	7	10	8.15	0.00	239.35	10
80	20	60.70	43.70	36	53	10	10	1.52	0.00	105.99	10
80	40	122.00	39.90	30	50	5	10	4.38	0.00	286.17	10
80	60	194.20	28.40	16	40	3	10	7.55	0.00	189.76	10
80	80	252.30	20.20	10	26	3	10	7.04	0.00	145.30	10
100	25	94.50	51.30	40	68	5	10	0.75	0.00	310.30	10
100	50	172.10	41.40	34	49	0	10	2.49	0.00	491.01	10
100	75	284.60	23.80	15	34	1	10	4.46	0.00	276.44	10
100	100	385.00	16.50	12	24	1	10	4.03	0.00	174.80	10
200	50	386.50	56.60	49	71	0	10	0.00	0.00	1943.35	10
200	100	796.10	15.30	11	23	0	10	0.89	0.00	330.36	10
200	150	1187.80	3.70	1	9	7	10	0.00	0.00	49.39	10
200	200	1602.20	1.00	0	3	10	10	0.00	0.00	6.59	10

32.4 min with the 200×50 tables but achieved the optimum for all instances. For this 10 table set the Cplex failed to solve all the integer problems due to memory limitations.

For the second set of instances we considered two weighting alternatives: $\xi_{ij} = a_{ij}$ and $\xi_{ij} = 1$, $(i,j) \in \mathscr{C} \setminus S_1$. When all cell weights are equal there are many different solutions with the same value and the search for an optimal solution is likely to be more arduous.

Table 3
Results on the second set of tables with $\xi_{ij} = a_{ij},\ (i,j)\ \in \mathscr{C} \setminus S_1$

$m+1$	$n+1$	p	up	minup	maxup	Cpl	Lag	IGap	FGap	T (s)	Cpl+Lag
50	10	24.00	22.80	20	24	10	10	0.00	0.00	6.64	10
50	20	49.00	36.80	28	45	10	10	0.85	0.00	49.74	10
50	30	73.00	39.40	20	64	10	9	0.81	0.09	235.58	10
50	40	98.00	34.60	27	43	10	9	0.21	0.19	439.80	10
50	50	122.00	27.90	22	48	9	9	0.03	0.03	374.65	9
60	20	58.00	43.10	33	54	10	9	0.16	0.16	136.37	10
60	40	117.00	33.70	23	42	9	10	0.88	0.00	301.67	10
60	60	174.50	24.50	19	31	7	10	0.00	0.00	35.31	10
80	20	77.00	45.60	32	65	10	10	0.13	0.00	91.10	10
80	40	155.30	31.10	22	40	6	10	0.00	0.00	28.24	10
80	60	232.90	21.10	15	29	5	10	2.00	0.00	22.88	10
80	80	311.80	16.90	10	24	5	8	19.96	19.96	112.04	9
100	20	97.00	45.40	36	55	8	10	0.00	0.00	42.05	10
100	40	195.20	35.20	26	40	0	10	0.00	0.00	61.91	10
100	60	292.80	20.40	17	29	2	10	0.48	0.00	47.42	10
100	80	390.10	14.20	10	20	3	10	0.00	0.00	57.30	10
100	100	491.10	9.00	3	12	6	8	19.98	19.98	138.59	9

Table 4
Results on the second set of tables with $\xi_{ij} = 1\ (i,j)\ \in \mathscr{C} \setminus S_1$

$m+1$	$n+1$	p	up	minup	maxup	Cpl	Lag	IGap	FGap	T (s)	Cpl+Lag
50	10	24.00	22.80	20	24	3	10	2.20	0.00	14.83	10
50	20	49.00	36.80	28	45	2	10	11.23	0.00	92.18	10
50	30	73.00	39.40	20	64	4	3	27.89	5.95	304.86	5
50	40	98.00	34.60	27	43	3	0	36.99	10.03	461.29	3
50	50	122.00	27.90	22	48	6	3	47.58	8.89	355.28	8
60	20	58.00	43.10	33	54	3	10	5.81	0.00	99.41	10
60	40	117.00	33.70	23	42	5	1	21.09	5.51	281.20	5
60	60	174.50	24.50	19	31	7	1	41.34	9.31	292.75	7
80	20	77.00	45.60	32	65	5	10	1.62	0.00	93.50	10
80	40	155.30	31.10	22	40	4	9	11.92	0.63	202.58	10
80	60	232.90	21.10	15	29	1	5	33.29	3.59	274.28	5
80	80	311.80	16.90	10	24	0	7	43.58	2.91	218.64	7
100	20	97.00	45.40	36	55	6	10	1.05	0.00	113.41	10
100	40	195.20	35.20	26	40	0	10	3.48	0.00	209.84	10
100	60	292.80	20.40	17	29	0	10	8.14	0.00	123.32	10
100	80	390.10	14.20	10	20	2	10	27.09	0.00	136.40	10
100	100	491.10	9.00	3	12	2	10	30.00	0.00	46.19	10

Table 3 reports the results obtained with $\xi_{ij} = a_{ij},\ (i,j) \in \mathscr{C} \setminus S_1$. The problem was solved to optimality in all instances. With Cplex 71% of the instances were solved. With the Lagrangian procedure 95.3% in-proven optima were obtained. Only 3 final incumbents were neither in-proven nor out-proven optima. As an attempt to find out whether or not they were optimal, the lower and upper bounds generated by the Lagrangian procedure were used to add two new constraints to formulation (P_N) and Cplex was set to run the resulting formulations. With the new constraints the memory limitations were overcome and all three incumbents were proved to be optimal.

Table 4 reports the computational results obtained for the same tables with $\xi_{ij} = 1,\ (i,j) \in \mathscr{C} \setminus S_1$. With the unitary costs criterion the proposed methodologies showed to be less effective as expected. The problem was solved to proven optimality in 140 out of the 170 instances. Only 53 integer problems were solved with Cplex. The Lagrangian approach produced 119 in-proven and 12 out-proven optima. All the nonoptimal initial incumbents were improved along the Lagrangian procedure. When comparing the optimal solutions obtained with Cplex with the incumbents without in-proven optimality we found out that there were nine proven nonoptimal incumbents: one with three more complementary suppressions than the optimal number and eight with this gap equal to one. For the set of 30 tables for which no guaranteed optimal solution was achieved we used the lower and upper bounds of the Lagrangian procedure to add different constraints to the integer formulations in search of more information on the incumbent solutions' quality. We concluded that 13 incumbents were optimal and that one incumbent had one more complementary suppression than the optimal. We were left with 12 incumbents with one more complementary suppression than the Lagrangian lower bound on their number, two incumbents with this gap equal to 2, one with gap equal to 3 and another one with gap equal to 4. As whenever the lower bound is not tight this gap must be at least one, we suspect that many of the 16 solutions without proven optimality would be out-proven optimal, had we been able to compute the corresponding integer optimum with Cplex.

6. Final remarks

In this work we addressed the exact disclosure problem with no constraints other than those resulting from the table structure. Some extra conditions, imposed by other authors, may easily be incorporated in our models. To impose that all subtotals must be published, as in [4], one may simply drop the variables associated with cells $(i, n+1),\ i \in R$, and $(m+1, j),\ j \in C$ and the corresponding constraints, reducing the model's dimensions. If no zero cells may be suppressed then formulation (P) may be used fixing some variables at zero as detailed in the previous section. This additional condition is mentioned in [5] for frequency count tables and is imposed by some authors in their computational experience.

If the values of some nonsensitive cells are considered especially important to the user community one may assign them very large weights ξ_{ij} avoiding their inclusion in S_2 unless no other alternative exists. If the weight modification is done cell by cell it gives an assessment of the extra loss of nonconfidential information due to the publication of the corresponding cell value.

Weights may also be altered for another purpose. The effective protection given by any feasible solution of (P_N) to a sensitive cell (i_k, j_k) is bounded from below by the value of the smallest positive suppressed cell in the path that links nodes i_k and j_k, defined either by the y^k or by the w^k variables. To generate a near optimal solution with a protection range for (i_k, j_k) with width at least equal to a given threshold τ one can assign very large weights to all cells with value smaller than τ in the shortest path computations for cell (i_k, j_k).

When traditional protection intervals of the form $[a_k - l_k, a_k + u_k]$ are defined for each sensitive cell k, more than one circuit of suppressions may be required to protect each cell, because it must be possible to send a flow of value u_k from j_k to i_k and a flow of value l_k from i_k to j_k in the solution support graph. Substituting the binary path variables for nonnegative flow variables it is easy to deduce from (P_N) a valid compact formulation for the traditional interval protection version of the cell suppression problem. However, as in this case the coupling constraints link binary and nonnegative variables, the LP bound is much weaker. To strengthen it two tracks are being tried. The first track retrieves the path approach in (P_N): to be able to send a positive flow from a source node to a sink node it is necessary to have a directed path with positive capacity from the source to the sink. Combining path variables with flow variables in the coupling constraints cuts off some poor solutions of the linear relaxation of the flow-based model. The second track is to derive lifted versions of the conditions on the x_{ij} variables proposed in [8] to act directly on the binary suppression variables in the combined flow-path model.

Acknowledgements

The authors thank two anonymous referees for their comments, which have helped in improving the paper.

Appendix A.

Consider the undirected bipartite graph $G=(V,\mathscr{E})$ defined at the end of Section 2. Define in G, p paths $\mathscr{C}^1,\mathscr{C}^2,\dots,\mathscr{C}^p$, such that $\mathscr{C}^k$ is a path linking nodes i_k and j_k that does not include edge (i_k,j_k), for $k=1,\dots,p$.

The minimum cost set of paths that yields a cycle for each sensitive edge (i_k,j_k) is an optimal solution of the following binary problem:

$$(U)\quad \min \sum_{(i,j)\in(R\times C)\setminus S_1} \xi_{ij}x_{ij}$$

s.t.

$$\text{for } k=1,\dots,p, \tag{A.0}$$

$$\sum_{i\in R} v^k_{ij_k}=1, \tag{A.1}$$

$$\sum_{j\in C} v^k_{i_kj}=1, \tag{A.2}$$

$$\sum_{j\in C} v^k_{ij}=2\alpha^k_i,\quad i\in R\setminus\{i_k\}, \tag{A.3}$$

$$\sum_{i\in R} v^k_{ij}=2\beta^k_j,\quad j\in C\setminus\{j_k\}, \tag{A.4}$$

$$v^k_{ij}\leqslant x_{ij},\quad (i,j)\in(R\times C)\setminus S_1, \tag{A.5}$$

$$\alpha^k_i\in\{0,1\},\quad i\in R\setminus\{i_k\}, \tag{A.6}$$

$$\beta^k_j\in\{0,1\},\quad j\in C\setminus\{j_k\}, \tag{A.7}$$

$$v_{ij}^k \in \{0,1\}, \quad (i,j) \in R \times C, \tag{A.8}$$

$$x_{ij} \in \{0,1\}, \quad (i,j) \in (R \times C) \setminus S_1, \tag{A.9}$$

where

$$\underset{(i,j)\in(R\times C)\setminus S_1}{x_{ij}} = \begin{cases} 1 & \text{if cell } (i,j) \text{ is suppressed,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\underset{\substack{(i,j)\neq(i_k,j_k)\\ i\in R, j\in C}}{v_{ij}^k} = \begin{cases} 1 & \text{if edge } (i,j) \text{ is included in path } \mathscr{C}^k, \\ 0 & \text{otherwise,} \end{cases}$$

$$\underset{i\in R\setminus\{i_k\}}{\alpha_i^k} = \begin{cases} 1 & \text{if node } i \text{ is included in path } \mathscr{C}^k, \\ 0 & \text{otherwise,} \end{cases}$$

$$\underset{j\in C\setminus\{j_k\}}{\beta_j^k} = \begin{cases} 1 & \text{if node } j \text{ is included in path } \mathscr{C}^k, \\ 0 & \text{otherwise.} \end{cases}$$

As G is a bipartite graph every path $\mathscr{C}^k$ must include at least one node in R, at least one node in C and at least three edges. If an edge is included in a path so are its end nodes. So, the following inequalities are valid for (U):

$$\sum_{i\in[R\setminus\{i_k\}]} \alpha_i^k \geqslant 1, \quad k=1,\dots,p, \tag{A.10}$$

$$\sum_{j\in[C\setminus\{j_k\}]} \beta_j^k \geqslant 1, \quad k=1,\dots,p, \tag{A.11}$$

$$\sum_{(i,j)\in[(R\times C)\setminus\{(i_k,j_k)\}]} v_{ij}^k \geqslant 3, \quad k=1,\dots,p, \tag{A.12}$$

$$v_{ij}^k \leqslant \alpha_i^k, \quad i\in R\setminus\{i_k\}, \quad j\in C, \; k=1,\dots,p, \tag{A.13}$$

$$v_{ij}^k \leqslant \beta_j^k, \; j\in C\setminus\{j_k\}, \quad i\in R, \; k=1,\dots,p. \tag{A.14}$$

As in any feasible solution of $(\bar{U})$, the LP relaxation of (U), the following equalities hold:

$$\sum_{i\in[R\setminus\{i_k\}]} \alpha_i^k = \sum_{j\in[C\setminus\{j_k\}]} \beta_j^k, \quad k=1,\dots,p,$$

inequalities (A.10), (A.11), (A.12) are equivalent in $(\bar{U})$. In $(\bar{U})$ any feasible solution $(\bar{X},\bar{V},\bar{\alpha},\bar{\beta})$ satisfying (A.13) and (A.14) also satisfies them.

Let (RU) be the binary problem defined by (A.0), (A.1)–(A.9), (A.13) and (A.14), and $(\overline{RU})$ its LP relaxation.

For $(\bar{P})$, the linear relaxation of directed formulation (P) defined in Section 3, there is always an optimal solution, say $(\bar{X},\bar{Y})$ that verifies the following conditions:

$$\bar{y}_{ij}^k \times \bar{y}_{ji}^k = 0, \quad (i,j)\in[(R\times C)\setminus\{(i_k,j_k)\}], k=1,\dots,p,$$

$$\sum_{j\in C}\bar{y}^k_{ji}=\sum_{j\in C}\bar{y}^k_{ij}\leqslant 1,\quad i\in R,\ k=1,\ldots,p,$$

$$\sum_{i\in R}\bar{y}^k_{ij}=\sum_{i\in R}\bar{y}^k_{ji}\leqslant 1,\quad j\in C,\ k=1,\ldots,p.$$

A feasible solution of $(\overline{RU})$ may then be built from $\bar{Y}$ setting

$$\bar{v}^k_{ij}=\bar{y}^k_{ij}+\bar{y}^k_{ji},\quad (i,j)\in(R\times C),\ k=1,\ldots,p,$$

$$\bar{\alpha}^k_i=\frac{1}{2}\sum_{j\in C}(\bar{y}^k_{ij}+\bar{y}^k_{ji}),\quad i\in R\setminus\{i_k\},\ k=1,\ldots,p,$$

$$\bar{\beta}^k_j=\frac{1}{2}\sum_{i\in R}(\bar{y}^k_{ij}+\bar{y}^k_{ji}),\quad j\in C\setminus\{j_k\},\ k=1,\ldots,p.$$

So the optimum of $(\bar{P})$ is always greater than or equal to the optimum of $(\overline{RU})$. For some tables this inequality is strict.

References

[1] Willenborg L, de Waal T. Elements of statistical disclosure control. Berlin: Springer-Verlag; 2001.

[2] Cox LH. Network models for complementary cell suppression. Journal of the American Statistical Association 1995;90(432):1453–62.

[3] Carvalho FD, Dellaert N, Osório M. Statistical disclosure in two-dimensional tables: positive tables. Report 9441/a, Econometric Institute, Erasmus University Rotterdam, The Netherlands, 1994.

[4] Carvalho FD, Dellaert N, Osório M. Statistical disclosure in two-dimensional tables: general tables. Journal of the American Statistical Association 1994;89:1547–57.

[5] Willenborg L, de Waal T. Statistical disclosure control in practice. Berlin: Springer-Verlag; 1996.

[6] Kelly J, Golden B, Assad A. Cell suppression: disclosure protection for sensitive tabular data. Networks 1992;22: 397–417.

[7] Fischetti M, Salazar JJ. Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control. Mathematical Programming 1999;84:283–312.

[8] Carvalho FD, Almeida MT. Lower-bounding procedures for the 2-dimensional cell suppression problem. European Journal of Operational Research 2000;123:29–41.

[9] Cox LH. Disclosure analysis and cell suppression. Proceedings of the American Statistical Association, Social Statistics Section, 1975. p. 380–2.

[10] Cox LH. Suppression methodology and statistical disclosure control. Journal of the American Statistical Association 75 (Theory and Methods Section) 1980;370:377–85.

[11] Gusfield D. A graph theoretic approach to statistical data security. SIAM Journal of Computing 1988;17(3):552–71.

[12] Ahuja RK, Magnanti TL, Orlin JB. Network flows-theory, algorithms and applications. Englewood Cliffs, NJ: Prentice Hall; 1993.

[13] Geoffrion AM. Lagrangean relaxation for integer programming. Mathematical Programming Study 1974;2:82–114.

[14] Almeida MT, Carvalho FD. Covering edges by cycles in a bipartite graph at minimum cost. Working Paper 9/2001, Centro de Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, 2001.

[15] Held M, Wolfe P, Crowder H. Validation of subgradient optimization. Mathematical Programming 1974;6:62–88.