Partial Cell Suppression: a New Methodology for Statistical Disclosure Control

Matteo Fischetti *(DEI, University of Padova, Italy)*
Juan-José Salazar-González *(DEIOC, University of La Laguna, Spain)*[*]

October, 1998

Summary. In this paper we address the problem of protecting confidentiality in statistical tables containing sensitive information that cannot be disseminated. This is an issue of primary importance in practice. Cell Suppression is a widely-used technique for avoiding disclosure of sensitive information, which consists in suppressing all sensitive table entries along with a certain number of other entries, called complementary suppressions. Determining a pattern of complementary suppressions that minimizes the overall loss of information results into a difficult (i.e., $\mathcal{NP}$-hard) optimization problem known as the Cell Suppression Problem. We propose here a different protection methodology consisting of replacing some table entries by appropriate intervals containing the actual value of the unpublished cells. We call this methodology Partial Cell Suppression, as opposed to the classical "complete" cell suppression. Partial cell suppression has the important advantage of reducing the overall information loss needed to protect the sensitive information. Also, the new method provides automatically auditing ranges for each unpublished cell, thus saving an often time-consuming task to the statistical office while increasing the information explicitly provided with the table. Moreover, we propose an efficient (i.e., polynomial-time) algorithm to find an optimal partial suppression solution. A preliminary computational comparison between partial and complete suppression methodologies is reported. Finally, we address possible extensions leading to a unified complete/partial cell suppression framework.

Keywords: Cell Suppression; Statistical Data Protection; Statistical Disclosure Control; Confidentiality; Linear Programming.

[*] *Address for correspondence*: Departamento de Estadística, I.O. y Computación, Facultad de Matemáticas, Universidad de La Laguna, 38271 La Laguna, Tenerife, Spain. E-mail: jjsalaza@ull.es

	A	B	C	Total
Activity I	20	50	10	80
Activity II	8	19	**22**	49
Activity III	17	32	12	61
Total	45	101	44	190

Figure 1: Investment of enterprises by activity and region.

1 Introduction

A statistical agency collects data to be processed and published. This data is often obtained from individual respondents under a pledge of confidentiality: statistical agencies have the responsibility of not releasing any data or data summaries from which individual respondent information can be revealed (*sensitive data*). On the other hand, statistical agencies aim at publishing as much information as possible. This results into a trade-off between privacy rights and information loss, an issue of primary importance in practice. The interested reader is referred, e.g., to Willenborg and Waal (1996) for a discussion on disclosure control methodologies.

Cell suppression is one of the most popular techniques for disclosure avoidance, and is typically applied to 2- or 3-dimensional tables whose entries (*cells*) are subject to marginal totals. We next outline the standard *complete* cell suppression technique, and then introduce an alternative version that we call *partial* cell suppression. Both methodologies are illustrated by means of the following introductory example, taken from Willenborg and Waal (1996).

Figure 1 exhibits a table giving the investment of enterprises (per millions of guilders), classified by activity and region. Let us assume that the information in the cell corresponding to Activity II and Region C is confidential, hence it is viewed as a *sensitive cell* to be suppressed. By using the marginal totals an *attacker* interested in the disclosure of the sensitive cell can however recompute its missing value, hence other table entries must be suppressed as well, e.g. those of Figure 2.

With this choice, the attacker cannot disclose the nominal value of the sensitive cell exactly, although he/she can still compute a range for the values of this cell which

	A	B	C	Total
Activity I	20	50	10	80
Activity II	*	19	*	49
Activity III	*	32	*	61
Total	45	101	44	190

Figure 2: A possible complete CSP output.

are consistent with the published entries. Indeed, the minimum value y_{23}^- for sensitive cell (2,3) can be computed by solving a linear program in which the values y_{ij} for the missing cells (i, j) are treated as unknowns, namely

$$y_{23}^- := \min y_{23}$$

subject to

$$\begin{array}{rrrrl} y_{21} & & +y_{23} & & = 30 \\ & y_{31} & & +y_{33} & = 29 \\ y_{21} & +y_{31} & & & = 25 \\ & & y_{23} & +y_{33} & = 34 \\ y_{21}, & y_{23}, & y_{31}, & y_{33} & \geq 0. \end{array}$$

Notice that the right-hand side values are known to the attacker, as they can be obtained as the difference between the marginal and the published values in a row/column.

The maximum value y_{23}^+ for the sensitive cell can be computed in a perfectly analogous way, by solving the linear program of maximizing y_{23} subject to the same constraints as before.

In the example, $y_{23}^- = 5$ and $y_{23}^+ = 30$, i.e., the sensitive information is "protected" within the *protection interval* [5,30]. If this interval is considered sufficiently wide by the statistical office, then the sensitive cell is called *protected*; otherwise new complementary suppressions are needed. Notice however that the extreme values of interval $[5, 30]$ are only attained if the cell corresponding to Activity II and Region A takes the quite unreasonable values of 0 and 25. Bounding the cell variation to $\pm 50\%$ of the nominal value (say) results in the more realistic protection interval $[18, 26]$.

The statistical office then aims at finding a set of complementary suppressions protecting all the sensitive cells against an attacker, and such that the loss of information associated with the suppressed entries is minimized.

This combinatorial optimization problem is known as the *(Complete) Cell Suppression Problem*, or CSP for short. CSP belongs to the class of the strongly $\mathcal{NP}$-hard problems (see, e.g., Kelly et al. (1992), Geurts (1992), Kao (1996), meaning that it is very unlikely that an algorithm for the exact solution of CSP exists, which guarantees an efficient (i.e., polynomial-time) performance for all possible input instances.

Previous work on CSP from the literature mainly concentrate on 2-dimensional tables with marginals. Heuristic solution procedures have been proposed by several authors, including Cox (1980, 1995), Sande (1984), Kelly et al. (1992), and Carvalho et al. (1994). Kelly (1990) proposed a mixed-integer linear programming formulation involving a huge number of variables and constraints. Geurts (1992) refined this model, and reported computational experiences on small-size instances, the largest instance solved to optimality being a table with 20 rows, 6 columns and 17 sensitive cells. Gusfield (1988) gave a polynomial algorithm for a special case of the problem.

	A	B	C	Total
Activity I	[18,24]	50	[6,12]	80
Activity II	[4,10]	19	[20,26]	49
Activity III	17	32	12	61
Total	45	101	44	190

Figure 3: A possible partial CSP output

Heuristics for 3-dimensional tables have been proposed in Robertson (1994), Sande (1984), and Dellaert and Luijten (1996).

Very recently, we proposed in Fischetti and Salazar (1998a) a new method capable of solving to proven optimality, on a personal computer, 2-dimensional tables with about 250,000 cells and 10,000 sensitive entries. An extension of this methodology resulted into an algorithm presented in Fischetti and Salazar (1998b), capable of solving to proven optimality real-world 3- and 4-dimensional tables as well as linked tables of reasonable size.

In this paper we address cell suppression in the very general context of tables whose entries are linked by a generic system of linear equations, as we proposed in Fischetti and Salazar (1998b). However, we introduce and analyze here an alternative protection methodology that we call *partial cell suppression*. The methodology consists in publishing intervals for some table entries, as opposed to replacing some table entries by "missing". The published intervals must provide a convenient set of possible values for the correspondent entries, containing the true original value. Figure 3 exhibits a possible partial cell suppression output for the original table in Figure 1, to be compared with the complete cell suppression output in Figure 2.

The new method has three important advantages over complete (i.e., standard) cell suppression. First, the loss of information needed to meet the required protection levels is typically significantly smaller by using the partial suppression methodology. A second advantage is related to computational complexity. Indeed, we provide in this paper an efficient (i.e., polynomial time) method for finding optimal partial suppressions. A third advantage is that the methodology itself provides, with no extra computational effort, the *auditing range* for each partially suppressed entry, defined by the minimum and maximum cell values that an attacker can compute by using the published information.

The paper is organized as follows. Section 2 gives a formal description of the new methodology (partial cell suppression), compared with the classical methodology (complete cell suppression). A Linear Programming model is presented and discussed in Section 3, whereas Section 4 outlines an efficient approach for solving such a model. Preliminary computational results are given in Section 5, where we compare the partial and complete cell suppression methodologies on test instances. Some conclusion

are finally drawn in Section 6, where we also discuss a unifying framework for partial and complete suppression.

A preliminary version of the present paper was presented by the authors at the *International Symposium on Linked Employer-Employee Data*, Washington, May 21–22, 1998.

2 Partial vs. Complete Cell Suppression

We first give a formal definition of the complete (i.e., yes/no) cell suppression problem, in the general form we introduced in Fischetti and Salazar (1998b).

A *table* is a vector $a = [a_1, \ldots, a_n]$ satisfying a given set of linear constraints known to a possible attacker, namely

$$My = b \tag{1}$$

$$lb_i \leq y_i \leq ub_i \quad \text{for all } i = 1, \ldots, n. \tag{2}$$

In other words, system (1)–(2) models the whole *a priori* information on the table known to an attacker, and $y := a$ is a possible solution. Typically, each equation in (1) corresponds to a marginal entry, whereas inequalities (2) enforce the public "external bounds". In the case of k-dimensional tables with marginals, each equation in (1) is of the type $\sum_{j \in Q_i} y_j - y_i = 0$, where index i corresponds to a marginal entry and Q_i to the associated set of internal table entries. Therefore, in this case M is a $\{0, +1, -1\}$-matrix and $b = 0$. Moreover, in case $k = 2$ the linear system (1)–(2) can be represented in a natural way as a *network*, a property having important theoretical and practical implications; see, e.g., Cox (1980, 1995), Kelly et al. (1992), and Fischetti and Salazar (1998a). Unfortunately, this nice structure is not preserved for $k \geq 3$, unless the table decomposes into a set of independent 2-dimensional subtables.

A *cell* is an index corresponding to an entry of the table.

Given a *nominal table* a, let $\{i_1, \ldots, i_p\}$ denote the set of sensitive cells in a to be protected, as identified by the statistical office according to some criteria. For each sensitive cell i_k $(k = 1, \ldots, p)$, the statistical office provides three nonnegative values, namely LPL_{i_k}, UPL_{i_k} and SPL_{i_k}, called *Lower Protection Level*, *Upper Protection Level* and *Sliding Protection Level*, respectively. In typical applications, these values are computed as a certain percentage of the nominal value a_{i_k}.

A *complete suppression pattern* is a subset of cells $SUP \subseteq \{1, \ldots, n\}$ corresponding to the unpublished cells. A *consistent table* with respect to a given suppression pattern SUP and to a given nominal table a is a vector $y = [y_1, \ldots, y_n]$ satisfying

$$My = b$$

$$lb_i \leq y_i \leq ub_i \qquad \text{for all } i \in SUP$$

$$y_i = a_i \qquad \text{for all } i \notin SUP$$

where the latter equations impose that the components of y associated with the published entries coincide with the nominal ones. In other words, any consistent table gives a feasible way the attacker can fill the missing entries of the published table.

A complete suppression pattern is considered *feasible* by the statistical office if it guarantees the required protection intervals against an attacker, in the sense that, for each sensitive cell i_k $(k = 1, \ldots, p)$, there exist two feasible tables, say f^k and g^k, such that $f^k_{i_k} \leq a_{i_k} - LPL_{i_k}$, $g^k_{i_k} \geq a_{i_k} + UPL_{i_k}$, and $g^k_{i_k} - f^k_{i_k} \geq SPL_{i_k}$. In other words, it is required that $y^-_{i_k} \leq a_{i_k} - LPL_{i_k}$, $y^+_{i_k} \geq a_{i_k} + UPL_{i_k}$, and $y^+_{i_k} - y^-_{i_k} \geq SPL_{i_k}$, where

$$\left.\begin{array}{rrcll} y^-_{i_k} := & \min y_{i_k} & & & \\ & My & = & b & \\ & lb_i \leq y_i & \leq & ub_i & \text{for all } i \in SUP \\ & y_i & = & a_i & \text{for all } i \notin SUP \end{array}\right\}$$

and

$$\left.\begin{array}{rrcll} y^+_{i_k} := & \max y_{i_k} & & & \\ & My & = & b & \\ & lb_i \leq y_i & \leq & ub_i & \text{for all } i \in SUP \\ & y_i & = & a_i & \text{for all } i \notin SUP \end{array}\right\}$$

As already mentioned, the statistical office is interested in selecting, among all complete suppression patterns, a feasible one with minimum information loss.

This issue can be modeled by associating a *weight* $w_i \geq 0$ with each cell i of the table, and by requiring the minimization of the overall weight of the suppressed cells, computed as $\sum_{i \in SUP} w_i$ (in typical applications, the weights w_i provided by the statistical office are proportional to a_i or to $\log a_i$). The resulting combinatorial problem will be denoted as the *Complete Cell Suppression Problem*, or complete CSP for short.

We next introduce the new cell suppression methodology, leading to what we call the *Partial CSP*. As already mentioned, the idea is to allow each table entry a_i to be replaced by a convenient interval $[a_i - z^-_i, a_i + z^+_i]$ (case $z^-_i = z^-_i = 0$ being allowed). The extreme values of each interval have then to be determined so as to ensure the required protection for the sensitive entries, while minimizing the overall loss of information incurred.

More specifically, assume that the statistical office is willing to publish, for each cell i, an interval $[a_i - z^-_i, a_i + z^+_i]$, where $a_i - z^-_i \geq lb_i$ and $a_i + z^+_i \leq ub_i$ for consistency with the *a priori* information known to the attacker (i.e., with the external bounds), and $z^-_i \geq 0$ and $z^+_i \geq 0$ so as to avoid publishing wrong information. Given such

intervals, an attacker can then compute all potential values for each sensitive cell i_k by solving the two LP's:

$$\left.\begin{array}{rrcll} y_{i_k}^- := & \min y_{i_k} & & & \\ & My & = & b & \\ & a_i - z_i^- \leq y_i & \leq & a_i + z_i^+ & \text{for all } i = 1, \ldots, n \end{array}\right\}$$

and

$$\left.\begin{array}{rrcll} y_{i_k}^+ := & \max y_{i_k} & & & \\ & My & = & b & \\ & a_i - z_i^- \leq y_i & \leq & a_i + z_i^+ & \text{for all } i = 1, \ldots, n \end{array}\right\}$$

As in the case of complete CSP, the statistical office is interested in defining the intervals $[a_i - z_i^-, a_i + z_i^+]$ so as to guarantee the required protection levels against a possible attacker. In other words, for each sensitive cell i_k we must have

$$\begin{array}{rl} \text{lower protection level requirement:} & y_{i_k}^- \leq a_{i_k} - LPL_{i_k} \\ \text{upper protection level requirement:} & y_{i_k}^+ \geq a_{i_k} + UPL_{i_k} \\ \text{sliding protection level requirement:} & y_{i_k}^+ - y_{i_k}^- \geq SPL_{i_k} \end{array}$$

In particular, this implies $z_{i_k}^+ \geq UPL_{i_k}$, $z_{i_k}^- \geq LPL_{i_k}$, and $z_{i_k}^+ + z_{i_k}^- \geq SPL_{i_k}$ for all sensitive cells i_k.

Moreover, the published intervals should be as narrow as possible so as to minimize the overall loss information, computed as

$$\sum_{i=1}^{n} (w_i^+ z_i^+ + w_i^- z_i^-)$$

for some pre-specified nonnegative weights w_i^+ and w_i^-. For example, setting $w_i^+ = w_i^- = 1$ for all i leads to the figure $\sum_{i=1}^{n}(z_i^+ + z_i^-)$, i.e., the overall loss of information equals to the sum of the interval widths.

An important observation is that the nonnegativity assumption on the weights implies minimality of the ranges $[a_i - \tilde{z}_i^-, a_i + \tilde{z}_i^+]$ associated with an optimal solution $(\tilde{z}^+, \tilde{z}^-)$, i.e., these ranges actually coincide with the *auditing ranges* $[y_i^-, y_i^+]$ an attacker can recompute, for each cell i, from the published information.

3 A Linear Programming model for partial CSP

We next introduce two Linear Programming (LP) models for the partial CSP defined in the previous section. The models are in the spirit of the Mixed-Integer LP model we gave in Fischetti and Salazar (1998b) for complete CSP, but they do not need any 0-1 variable to model the yes/no nature of a complete suppression pattern.

For notational convenience, let us define the relative external bounds:

$$LB_i := a_i - lb_i \geq 0$$

and

$$UB_i := ub_i - a_i \geq 0,$$

i.e., the *a priori* range of feasible values for cell i known to the attacker is $[a_i - LB_i, a_i + UB_i]$. We also define

$$LPL_i := UPL_i := SPL_i := 0 \quad \text{for all nonsensitive cells } i,$$

and assume, for consistency reasons, that $UPL_i \leq UB_i$, $LPL_i \leq LB_i$, and $SPL_i \leq LB_i + UB_i$ for all i.

3.1 A first model

Our first LP model for partial CSP requires the introduction, for each $k = 1, \ldots, p$, of auxiliary continuous variables $f^k = [f_1^k, \ldots, f_n^k]$ and $g^k = [g_1^k, \ldots, g_n^k]$ defining tables which are consistent with the published intervals, and certificating the fulfillment of the protection level requirements. The model then reads:

$$\min \sum_{i=1}^{n} (w_i^+ z_i^+ + w_i^- z_i^-) \tag{3}$$

subject to

$$UPL_i \leq z_i^+ \leq UB_i \qquad \text{for } i = 1, \ldots, n \tag{4}$$

$$LPL_i \leq z_i^- \leq LB_i \qquad \text{for } i = 1, \ldots, n \tag{5}$$

$$SPL_{i_k} \leq z_{i_k}^+ + z_{i_k}^- \qquad \text{for } k = 1, \ldots, p \tag{6}$$

and, for each sensitive cell i_k $(k = 1, \ldots, p)$:

$$\left.\begin{array}{rlll} M & f^k & = b & \\ a_i - z_i^- \leq & f_i^k & \leq a_i + z_i^+ & \text{for } i = 1, \ldots, n \\ & f_{i_k}^k & = a_{i_k} - z_{i_k}^- & \end{array}\right\} \tag{7}$$

$$\left.\begin{array}{rlll} M & g^k & = b & \\ a_i - z_i^- \leq & g_i^k & \leq a_i + z_i^+ & \text{for } i = 1, \ldots, n \\ & g_{i_k}^k & = a_{i_k} + z_{i_k}^+ & \end{array}\right\} \tag{8}$$

Constraints (4)–(6) express the consistency of the published intervals $[a_i - z_i^-, a_i + z_i^+]$ with the external bounds known to an attacker, and also enforce the protection

levels. Constraints (7) and (8) require both extreme values of the published intervals for sensitive cells be attained in some tables which are consistent with the published one. Similar constraints for nonsensitive cells could be considered as well, but they are redundant under the weight nonnegativity assumption.

Model (3)–(8) involves a really huge number of auxiliary variables f_i^k and g_i^k and of linking constraints between the (z^+, z^-) and the auxiliary variables. For example, for a 100×100 table with marginals having 5% sensitive cells the model would need more than 10,000,000 variables and 20,000,000 constraints, a size that cannot be handled explicitly by the today LP technology.

We next derive an equivalent model inspired by Benders' decomposition (see e.g. Nemhauser and Wolsey (1988)), which is based on the same technique we used in Fischetti and Salazar (1998b) in the context of complete CSP. The idea is to use standard LP duality theory to avoid the explicit introduction of the auxiliary variables f^k and g^k $(k = 1, \ldots, p)$ along with the associated linking constraints. In the new model, the protection level requirements are in fact imposed by means of families of linear constraints in the space of the (z^+, z^-)-variables only. Before formulating the new model, we need the following characterization of the vectors (z^+, z^-) for which systems (7) and (8) admit feasible solutions f^k and g^k, respectively.

3.2 Imposing the feasibility of system (8)

Assume that (z^+, z^-) is a given (arbitrary but fixed) parameter, and consider any given sensitive cell i_k $(k = 1, \ldots, p)$. Clearly, linear system (8) admits a feasible solution g^k if and only if

$$a_{i_k} + z^+_{i_k} \leq y^+_{i_k},$$

where $y^+_{i_k}$ is the optimal value of the linear program

$$y^+_{i_k} := \max y_{i_k} \tag{9}$$

subject to

$$My = b \tag{10}$$

$$y_i \leq a_i + z_i^+ \qquad \text{for all } i = 1, \ldots, n \tag{11}$$

$$-y_i \leq -a_i + z_i^- \qquad \text{for all } i = 1, \ldots, n. \tag{12}$$

This is a parametric LP problem in the y-variables only, with variable upper/lower bounds depending on the given parameter (z^+, z^-). We call (9)–(12) the *attacker subproblem* associated with the upper protection of sensitive cell i_k, with respect to parameter (z^+, z^-). By LP duality, this subproblem is equivalent to the linear

program:

$$y_{i_k}^{+} = \min \gamma^t\, b + \sum_{i=1}^{n} [\alpha_i\, (a_i + z_i^{+}) - \beta_i\, (a_i - z_i^{-})] \tag{13}$$

subject to

$$\left.\begin{array}{c} \alpha^t - \beta^t + \gamma^t M = e_{i_k}^t \\ \alpha \geq 0,\ \beta \geq 0,\ \gamma \text{ unrestricted in sign} \end{array}\right\} \tag{14}$$

where e_{i_k} denotes the i_k-th column of the identity matrix of order n, and γ, α and β are LP dual vectors associated with constraints (10), (11) and (12), respectively.

It then follows that the linear system (8) has a feasible solution if and only if

$$a_{i_k} + z_{i_k}^{+} \leq y_{i_k}^{+} = \min \left\{ \gamma^t\, b + \sum_{i=1}^{n} [\alpha_i\, (a_i + z_i^{+}) - \beta_i\, (a_i - z_i^{-})] : (14) \text{ holds} \right\},$$

i.e., if and only if

$$a_{i_k} + z_{i_k}^{+} \leq \gamma^t\, b + \sum_{i=1}^{n} [\alpha_i\, (a_i + z_i^{+}) - \beta_i\, (a_i - z_i^{-})] \quad \text{for all } \alpha, \beta, \gamma \text{ satisfying (14).}$$

Because of (14) and $Ma = b$, we have $\gamma^t b + \sum_{i=1}^{n} (\alpha_i a_i - \beta_i a_i) = \gamma^t M a + (\alpha - \beta)^t a = e_{i_k}^t a = a_{i_k}$. Hence the above system can be rewritten as

$$\sum_{i=1}^{n} (\alpha_i z_i^{+} + \beta_i z_i^{-}) \geq z_{i_k}^{+} \quad \text{for all } \alpha, \beta, \gamma \text{ satisfying (14).} \tag{15}$$

In other words, system (15) defines a set of constraints in the (z^+, z^-)-space which is equivalent to the feasibility of system (8).

Notice that (15) contains in principle an infinite number of constraints, each associated with a different point (α, β, γ) of the polyhedron defined by (14). However, it is well known that only the extreme points (and rays) of such polyhedron can lead to undominated constraints (15), i.e., a finite number of such constraints is sufficient to impose the feasibility of system (8) for a given sensitive cell i_k.

3.3 Imposing the feasibility of system (7)

Analogously, imposing the existence of a feasible solution f^k of the linear system (7) for a given cell i_k is equivalent to imposing $y_{i_k}^{-} \leq a_{i_k} - z_{i_k}^{-}$, where

$$y_{i_k}^{-} := \min\{y_{i_k} : (10)\text{–}(12) \text{ hold } \} \equiv -\max\{-y_{i_k} : (10)\text{–}(12) \text{ hold } \}. \tag{16}$$

This is called the *attacker subproblem* associated to the lower protection of sensitive cell i_k, with respect to the given parameter (z^+, z^-). By LP duality, this subproblem is equivalent to the linear program

$$y^-_{i_k} = \min \gamma^t\, b + \sum_{i=1}^{n} [\alpha_i\,(a_i + z^+_i) - \beta_i\,(a_i - z^-_i)] \tag{17}$$

subject to

$$\left.\begin{array}{r} \alpha^t - \beta^t + \gamma^t M = -e^t_{i_k} \\ \alpha \geq 0,\ \beta \geq 0,\ \gamma \text{ unrestricted in sign.} \end{array}\right\} \tag{18}$$

Hence the feasibility of system (7) for cell i_k can be restated as

$$a_{i_k} - z^-_{i_k} \geq -\left(\gamma^t\, b + \sum_{i=1}^{n} [\alpha_i\,(a_i + z^+_i) - \beta_i\,(a_i - z^-_i)]\right) \quad \text{for all } \alpha, \beta, \gamma \text{ satisfying (18)}$$

or, equivalently,

$$\sum_{i=1}^{n} (\alpha_i z^+_i + \beta_i z^-_i) \geq z^-_{i_k} \quad \text{for all } \alpha, \beta, \gamma \text{ satisfying (18).}$$

3.4 A second model

The above characterization of the feasible vectors (z^+, z^-) leads to the following new LP model for partial CSP:

$$\min \sum_{i=1}^{n} (w^+_i z^+_i + w^-_i z^-_i) \tag{19}$$

subject to

$$UPL_i \leq z^+_i \leq UB_i \qquad \text{for } i = 1, \ldots, n \tag{20}$$

$$LPL_i \leq z^-_i \leq LB_i \qquad \text{for } i = 1, \ldots, n \tag{21}$$

$$SPL_{i_k} \leq z^-_{i_k} + z^-_{i_k} \qquad \text{for } k = 1, \ldots, p \tag{22}$$

and, for each sensitive cell i_k $(k = 1, \ldots, p)$:

$$\left.\begin{array}{r} \sum_{i=1}^{n} (\alpha_i z^+_i + \beta_i z^-_i) \geq z^+_{i_k} \\ \text{for all extreme points } (\alpha, \beta, \gamma) \text{ satisfying (14)} \end{array}\right\} \tag{23}$$

$$\left.\begin{array}{r} \sum_{i=1}^{n} (\alpha_i z^+_i + \beta_i z^-_i) \geq z^-_{i_k} \\ \text{for all extreme points } (\alpha, \beta, \gamma) \text{ satisfying (18)} \end{array}\right\} \tag{24}$$

Notice that, for a given sensitive cell i_k, constraints (23) need to be imposed explicitly only if $UPL_{i_k} > 0$, or $UPL_{i_k} = 0$ and $SPL_{i_k} > LPL_{i_k}$. Analogously, constraints (24) are only needed when $LPL_{i_k} > 0$, or $LPL_{i_k} = 0$ and $SPL_{i_k} > UPL_{i_k}$.

We call (23)–(24) the *capacity constraints* in analogy with similar constraints we introduced in Fischetti and Salazar (1998b) for the complete CSP. Intuitively, these capacity constraints force to publish sufficiently wide intervals for certain cells whose positions within the table and contributions to the overall protection are determined by the dual variables (α, β, γ) of the attacker subproblems.

4 Solving the two models

The solution of our first LP model (3)–(8) can in principle be obtained in polynomial time, as it involves a number of variable and constraints which is (huge but) polynomially bounded in the input size. In practice, however, the model cannot be handled effectively, even for very small instances.

At first glance, the LP relaxation of the second model (19)–(24) seems even more difficult to solve due to the exponential number of capacity constraints needed. In fact, in the second model the reduction on the number of variables was obtained through the introduction of a really huge number of inequalities, which cannot be handled explicitly by any LP solver. However, for a given partial CSP instance just a few capacity constraints are typically needed in the LP relaxation to force the protection level requirements. Although we cannot decide in advance which capacity constraints are the relevant ones, we can apply a dynamic constraint-generation technique that identifies them "on the fly", during the solution of a relaxed LP problem.

To be more specific, we approach the solution of the second model (19)–(24) through the following iterative *cutting-plane* scheme.

4.1 Iterative scheme

We start by solving the so-called *master LP*:

$$\min\{\sum_{i=1}^{n}(w_i^+ z_i^+ + w_i^- z_i^-) : \ (4)\text{–}(6) \text{ hold}\},$$

in which we only impose the trivial consistency conditions on the z_i^+ and z_i^- variables. Let $(\tilde{z}^+, \tilde{z}^-)$ be the optimal solution found. Our order of business it to check whether the vector $(\tilde{z}^+, \tilde{z}^-)$, now viewed as a given parameter, guarantees the required protection levels. In geometrical terms, this is equivalent to find an hyperplane in the

(z^+, z^-)-space that separates $(\tilde{z}^+, \tilde{z}^-)$ from the polyhedron defined by the capacity constraints. This is called the *separation problem* associated with the family of capacity constraints (23)–(24), and can be solved efficiently as follows.

For each sensitive cell i_k, in turn, we apply the following steps:

1. We first solve the attacker subproblem (9)–(12) for $z^+ = \tilde{z}^+$ and $z^- = \tilde{z}^-$, and check whether $a_{i_k} + \tilde{z}^+_{i_k} \leq y^+_{i_k}$. If this is the case, then $(\tilde{z}^+, \tilde{z}^-)$ satisfies the upper protection level requirement for the given i_k, hence all the exponentially many capacity constraints (23) are certainly fulfilled. Otherwise, the optimal dual solution $(\alpha^+, \beta^+, \gamma^+)$ of the attacker subproblem satisfies (14) and

$$\gamma^{+t}\, b + \sum_{i=1}^{n} [\alpha_i^+ \,(a_i + \tilde{z}_i^+) - \beta_i^+ \,(a_i - \tilde{z}_i^-)] = y_{i_k}^+ < a_{i_k} + z_{i_k}^+,$$

   hence it induces a capacity constraint $\sum_{i=1}^{n} (\alpha_i^+ z_i^+ + \beta_i^+ z_i^-) \geq z_{i_k}^+$ in family (23) that is violated by $(\tilde{z}^+, \tilde{z}^-)$. This constraint is then added to the master LP.

2. We then check whether $(\tilde{z}^+, \tilde{z}^-)$ satisfies the lower protection level requirement for i_k, which requires the solution of the attacker subproblem (16) associated with the lower protection level of cell i_k, for $z^+ = \tilde{z}^+$ and $z^- = \tilde{z}^-$. If this is not the case, then an optimal dual solution $(\alpha^-, \beta^-, \gamma^-)$ of the attacker subproblem induces a capacity constraint $\sum_{i=1}^{n} (\alpha_i^- z_i^+ + \beta_i^- z_i^-) \geq z_{i_k}^-$ in family (24) that is violated by $(\tilde{z}^+, \tilde{z}^-)$, which is then added to the master LP.

Clearly, Step 1 (respectively, Step 2) can be avoided if $UPL_{i_k} = SPL_{i_k} = 0$ (respectively, $LPL_{i_k} = SPL_{i_k} = 0$).

After having considered all sensitive cells i_k we have two possible cases. If no capacity constraint has been added to the master LP, then all of them are satisfied by $(\tilde{z}^+, \tilde{z}^-)$ which is therefore an optimal solution of the LP model (19)–(24). Otherwise, the master LP amended by the new capacity constraints is reoptimized, and the approach is iterated on the optimal solution $(\tilde{z}^+, \tilde{z}^-)$ of the new master LP.

By using the above cutting-plane scheme one can solve our second LP model quite efficiently. Actually, it can be shown that this approach can be implemented to run in polynomial time; see Nemhauser and Wolsey (1988) for a discussion about the equivalence of optimization and separation.

4.2 Constraint management

A crucial ingredient of our cutting-plane scheme is the run-time generation of constraints to be added to the master LP's. Although we generate just a few capacity constraints at each call of the separation procedures, the number of constraints produced during the overall solution scheme can become very large. On the other hand,

not all the generated constraints are active at the same time, hence one can think of a dynamic mechanism to drop slack constraints from the master LP at a certain iteration, and possibly to reinsert them at a later time. This is achieved through a data-structure called the *constraint pool* which stores all the constraints that have been generated by the separation procedure. After the solution of each master LP, a clean-up procedure removes from the LP (but not from the pool) the constraints that are slack at the optimal solution $(\tilde{z}^+, \tilde{z}^-)$. However, the removed constraints can turn out to be violated by the LP master solution $(\tilde{z}^+, \tilde{z}^-)$ at a later time. Therefore, before applying the separation phase we check whether the pool contains already violated constraints to be added to the current master LP. Only when no violated constraints are found in the pool, the separation procedures are applied to hopefully find new violated constraints to be inserted both in the pool and in the master LP.

In this way one can effectively manage a very large number of constraints, while keeping the master LP's to be solved at a reasonable size.

4.3 Problem reduction

The computing time spent in the solution of a given instance depends on its size, expressed in terms of both the number of decision variables involved and the number of nonzero protection levels (recall that zero protection levels do not induce capacity constraints). We next outline simple criteria to reduce the size of a given partial CSP instance (see Fischetti and Salazar (1998b) for a similar idea for the complete CSP).

A typically substantial reduction on the number of nonzero protection levels is achieved in a *preprocessing* phase, to be applied before entering the solution scheme. This is based on the observation that the trivial intervals $[a_i - LPL_i, a_i + UPL_i]$ alone may be already sufficient to protect some of the sensitive cells. To be specific, we consider the initial trivial solution $z^+ := UPL$ and $z^- := LPL$, and for each sensitive cell i_k we solve the attacker subproblem (9)–(12). In case $y^+_{i_k} \geq a_{i_k} + UPL_{i_k}$ one can clearly set $UPL_{i_k} := 0$, thus deactivating the upper protection level requirement for i_k. Otherwise, the capacity constraint associated with the dual optimal solution of the attacker subproblem qualifies as a relevant constraint, hence it is stored in the constraint pool data structure. A similar reasoning applies to lower and sliding protection levels.

A naive implementation of the above idea may be excessively time consuming for large instances, in that it may require even more computing time than the whole solution algorithm applied on the original instance. Hence a parametric resolution of the several attacker subproblems involved in preprocessing is needed. We suggest the following approach.

We introduce two p-dimensional arrays HIGH and LOW, whose entries HIGH_k and LOW_k give, respectively, lower and upper bounds on the solution value $y^+_{i_k}$ and

$y^-_{i_k}$ of the attacker subproblems associated with sensitive cell i_k, with respect to our initial trivial solution $[a - LPL, a + UPL]$. We initialize HIGH_k:=$\text{LOW}_k := a_{i_k}$ for all $k = 1, \ldots, p$, and then consider the sensitive cells i_k according to a non-increasing sequence of the associated values $\max\{SPL_{i_k}, UPL_{i_k} + LPL_{i_k}\}$. For each i_k, we first check whether $\text{HIGH}_k < a_{i_k} + UPL_{i_k}$ or $\text{HIGH}_k - \text{LOW}_k < SPL_{i_k}$, in which case we solve the attacker subproblem (9)–(12) with respect to the trivial solution $(z^+, z^-) = (UPL, LPL)$, obtaining a consistent table y^+ maximizing y_{i_k}. The entries of y^+ are then used to update all the entries of HIGH and LOW, by setting $\text{HIGH}_h = \max\{\text{HIGH}_h, y^+_{i_h}\}$ and $\text{LOW}_h = \min\{\text{LOW}_h, y^+_{i_h}\}$ for all $h = 1, \ldots, p$. We then check whether $\text{LOW}_k > a_{i_k} - LPL_{i_k}$ and $\text{HIGH}_k - \text{LOW}_k < SPL_{i_k}$, in which case we solve the attacker subproblem (16) and obtain a consistent table y^- minimizing y_{i_k}. As before, the entries of y^- are used to update all the entries of HIGH and LOW. Finally, we use the updated values HIGH_k and LOW_k to set $UPL_{i_k} = 0$ (if $\text{HIGH}_k \geq a_{i_k} + UPL_{i_k}$), $LPL_{i_k} = 0$ (if $\text{LOW}_k \leq a_{i_k} - LPL_{i_k}$), and $SPL_{i_k} = 0$ (if $\text{HIGH}_k - \text{LOW}_k \geq SPL_{i_k}$). In this way we avoid a typically substantial number of attacker subproblem resolutions.

Whenever a protection level associated with i_k is not satisfied, we have at hand a capacity constraint associated with the dual optimal solution of the corresponding attacker subproblem, which can be used to initialize the constraint pool. In other words, with no extra computing time we perform both the preprocessing phase and initialize the constraint pool with a number of relevant constraints.

As to a possible reduction of the number of variables in the master LP's to be solved, we have implemented a *variable-pricing* technique to speed-up the overall computation and to drastically reduce memory requirement when instances with more than 10,000 variables are considered; see, e.g. Nemhauser and Wolsey (1988) for details.

5 Preliminary computational results

A preliminary implementation of a solution algorithm for our second model has been obtained along the following lines. We eliminated the two decision vectors $z^+, z^- \in \mathbb{R}^n$ by introducing a single decision vector $x \in \mathbb{R}^n$ such that

$$\left.\begin{array}{c} z^+_i = UB_i\, x_i \\ z^-_i = LB_i\, x_i \\ 0 \leq x_i \leq 1 \end{array}\right\} \qquad i = 1, \ldots, n.$$

This is equivalent to force variables z^+_i and z^-_i to vary according to the prefixed ratio given by $LB_i z^+_i = UB_i z^-_i$. As a consequence, the new model has fewer x-variables, but its optimal solution value can be strictly worse than in the original model.

name	type	cells	links	p	pl
CBS1	41×31	1271	72	3	6
CBS2	183×61	11163	244	2467	4934
CCSR	359×46	16514	405	4923	9846

Table 1: Real-world instances.

The resulting simplified model was then solved by modifying slightly the code we presented in Fischetti and Salazar (1998b) for complete CSP. In fact, it was sufficient to deactivate all the "model improvements" related to the integrality of the x-variables, as described in Section 4.2 of Fischetti and Salazar (1998b), and to stop code execution right after the processing of the root node of the enumerative tree.

The modified software was compiled with Watcom C/C++ 10.6 and run on a PC Pentium/133 with 32 RAM Mbytes under Windows 95. As to the LP-solver, we used the commercial package CPLEX 3.0. Our preliminary test bed consists of the 2-dimensional real-world instances addresses in Fischetti and Salazar (1998b).

Table 1 reports information about the test bed. For each instance, the table gives:

name: name of the instance;

type: table size;

cells: number of cells in the table (=n);

links: number of equations in the table (=number of rows of matrix M);

p: number of sensitive cells;

pl: number of nonzero protection levels before problem reduction;

As to the critical values LB_i and UB_i defining the *a priori* intervals known to an attacker, we set:

$$UB_i := a_i \quad \text{and} \quad LB_i := a_i/2 \qquad \text{for each } i = 1, \ldots, n,$$

i.e., we assumed each table entry i is known with $[+100\%, -50\%]$ uncertainty. Protection levels were set to:

$$UPL_i := a_i/2 \quad \text{and} \quad LPL_i := a_i/4 \qquad \text{for each } i = 1, \ldots, n,$$

thus requiring 75% protection levels for the sensitive entries.

All fractional input values were rounded up to the nearest integer.

name	sup	loss	time	sup'	loss'	time'	time"
CBS1	9	153	4.78	7	486	367.62	0.07
CBS2	6	140,297	17.28	3	281,398	11.42	667.06
CCSR	85	9,357	77.67	38	19,099	116.93	2486.07

Table 2: Partial vs. complete suppression.

For each cell i, the partial suppression weights w_i^+ and w_i^- have been set to 1, whereas the weight for complete suppression has been defined as $w_i = UB_i + LB_i$, meaning that any completely suppressed cell introduces a loss of information equal to the width of the *a priori* interval known to the attacker for the missing value a_i. This produces complete CSP instances which can fairly be compared with their corresponding partial CSP versions.

Table 2 gives the following information.

name: name of the instance;

sup: number of partial suppressions in an optimal partial CSP solution;

loss: total loss of information in an optimal partial CSP solution;

time: overall computing time spent by our partial CSP approach (in Pentium/75 seconds);

sup': number of complete suppressions in an optimal complete CSP solution;

loss': total loss of information in an optimal complete CSP solution;

time': overall computing time spent by our complete CSP approach (in Pentium/75 seconds);

time": additional computing time needed to compute the auditing intervals $[y_i^-, y_i^+]$ for the unpublished cells in the optimal complete CSP solution.

The table clearly shows the information-loss savings obtainable by using the partial CSP methodology, even in the simplified version we implemented.

6 Conclusions and extensions

Cell suppression is a widely-used methodology in statistical disclosure control. In this paper we have introduced a different protection methodology consisting of replacing some table entries by appropriate intervals containing the actual value of the unpublished cells.

The new methodology, called partial cell suppression, has important advantages with respect to the classical one. First of all, it allows the statistical office to obtain the required protection levels with significantly smaller loss of information. In addition, the computational complexity of the new methodology is just polynomial as opposed to the exponential complexity of the classical cell suppression. Finally, the new method provides automatically auditing ranges for each unpublished cell, thus saving an often time-consuming task to the statistical office while increasing the information explicitly provided with the table.

We have presented a mathematical model and an efficient approach to solve it. We have also presented some preliminary computational results on a slight modification of our model.

Future work on the subject should address the issue of avoiding too many entries being replaced by intervals, even if this would produce a small increase of the overall information loss incurred. A possible solution is to associate a fixed penalty $w_i \geq 0$ to be applied whenever a table entry a_i is replaced by an interval, no matter how narrow the internal is. This requires the introduction in our two models of 0-1 variables x_i $(i = 1, \ldots, n)$, where $x_i = 1$ means that the entry a_i is replaced by an interval $[a_i - z_i^-, a_i + z_i^+]$ with $z_i^+ + z_i^- > 0$. In this way, the objective function become

$$\sum_{i=1}^{n} (w_i^+ z_i^+ + w_i^- z_i^- + w_i x_i)$$

and can incorporate the yes/no suppression penalties.

Of course, the new variables have to be linked to the continuous variables (z^+, z^-) by the consistency constraints:

$$\left.\begin{array}{c} z_i^+ \leq UB_i \; x_i \\ z_i^- \leq LB_i \; x_i \\ x_i \in \{0,1\} \end{array}\right\} \qquad \text{for } i = 1, \ldots, n.$$

which enforce $x_i = 1$ whenever $z_i^+ > 0$ and/or $z_i^- > 0$, i.e., when a proper interval is published instead of a_i. Notice that, in our second model, the above constraints can dealt with at the master level only, hence they do not affect capacity constraint separation. Also to be incorporated at the master level are possible constraints of the type

$$\sum_{i=1}^{n} x_i \leq S_{max},$$

imposing a maximum number S_{max} of "partially suppressed" entries, or constraints of the form

$$z_i^+ + z_i^- \geq W_i^{min} \; x_i,$$

imposing a minimum width W_i^{min} for each partially suppressed entry a_i.

The above extended model then generalizes and unifies both complete and partial cell suppression, and allows the user to have a complete control on the number of entries to be replaced by intervals as well as on the interval widths.

The exact/heuristic solution of the resulting $\mathcal{NP}$-hard optimization problem can be obtained by using enumeration techniques akin to the *branch-and-cut* method we described in Fischetti and Salazar (1998b) for complete cell suppression.

7 Acknowledgment

The first author was supported by *Ministero della Ricerca Scientifica e Tecnologica*, Italy, while the second author was supported by *Ministerio de Educacion y Ciencia*, Spain.

We thank Larry Cox (*U.S. Environment Protection Agency*, U.S.A.), Ramesh Dandekar (*U.S. Department of Energy*, U.S.A.), and Leon Willenborg (*Central Bureau of Statistics*, The Netherlands) for stimulating discussions on statistical disclosure control methodologies.

References

Cox, L. H. (1980) Suppression Methodology and Statistical Disclosure Control. *Journal of the American Statistical Association*, **75**, 377–385.

———(1995) Network Models for Complementary Cell Suppression. *Journal of the American Statistical Association*, **90**, 1453–1462.

Crowder, H. P., Johnson, E. L. and Padberg, M. W. (1983) Solving Large-Scale Zero-One Linear Programming Problems. *Operations Research*, **31**, 803–834.

Carvalho, F. D., Dellaert, N. P. and Osório, M. S. (1994) Statistical Disclosure in Two-Dimensional Tables: General Tables. *Journal of the American Statistical Association*, **89**, 1547–1557.

Dellaert, N. P. and Luijten, W. A. (1996) Statistical Disclosure in General Three-Dimensional Tables. Technical Paper TI 96-114/9, Tinbergen Institute.

Fischetti, M. and Salazar, J. J. (1998a) Models and Algorithms for the 2-Dimensional Cell Suppression Problem in Statistical Disclosure Control. *Mathematical Programming*, in the press.

———(1998b) Models and Algorithms for Optimizing Cell Suppression Problem in Tabular Data with Linear Constraints. Technical Paper, University of La Laguna, Tenerife.

Geurts, J. (1992) Heuristics for Cell Suppression in Tables. Technical Paper, Netherlands Central Bureau of Statistics, Voorburg.

Gusfield, D. (1988) A Graph Theoretic Approach to Statistical Data Security. *SIAM Journal on Computing*, **17**, 552–571.

Kao, M. Y. (1996) Data Security Equals Graph Connectivity. *SIAM Journal on Discrete Mathematics*, **9**, 87–100.

Kelly, J. P. (1990) Confidentiality Protection in Two and Three-Dimensional Tables. Ph.D. dissertation, University of Maryland, College Park, Maryland.

Kelly, J. P., Golden, B. L. and Assad, A. A. (1992) Cell Suppression: Disclosure Protection for Sensitive Tabular Data. *Networks* **22**, 397–417.

Nemhauser, G. L. and Wolsey, L. A. (1988) *Integer and Combinatorial Optimization.* New York: John Wiley & Sons.

Robertson, D. A. (1994) Cell Suppression at Statistics Canada. Proceedings of the *Second International Conference on Statistical Confidentiality*, Luxembourg.

Sande, G. (1984) Automated cell suppression to preserve confidentiality of business statistics. *Statistical Journal of the United Nations ECE* **2**, 33–41.

Willenborg, L. C. R. J. and Waal, T. de (1996) *Statistical Disclosure Control in Practice. Lecture Notes in Statistics 111.* New York: Springer.