Revisiting Interval Protection, a.k.a. Partial Cell Suppression, for Tabular Data

Jordi Castro[1(✉)] and Anna Via[2]

[1] Department of Statistics and Operations Research,
Universitat Politècnica de Catalunya,
Jordi Girona 1–3, 08034 Barcelona, Catalonia, Spain
jordi.castro@upc.edu

[2] School of Mathematics and Statistics, Universitat Politècnica de Catalunya,
Pau Gargallo 5, 08028 Barcelona, Catalonia, Spain
annaa35@gmail.com

Abstract. Interval protection or partial cell suppression was introduced in "M. Fischetti, J.-J. Salazar, Partial cell suppression: A new methodology for statistical disclosure control, *Statistics and Computing*, 13, 13–21, 2003" as a "linearization" of the difficult cell suppression problem. Interval protection replaces some cells by intervals containing the original cell value, unlike in cell suppression where the values are suppressed. Although the resulting optimization problem is still huge—as in cell suppression, it is linear, thus allowing the application of efficient procedures. In this work we present preliminary results with a prototype implementation of Benders decomposition for interval protection. Although the above seminal publication about partial cell suppression applied a similar methodology, our approach differs in two aspects: (i) the boundaries of the intervals are completely independent in our implementation, whereas the one of 2003 solved a simpler variant where boundaries must satisfy a certain ratio; (ii) our prototype is applied to a set of seven general and hierarchical tables, whereas only three two-dimensional tables were solved with the implementation of 2003.

Keywords: Statistical disclosure control · Tabular data · Interval protection · Cell suppression · Linear optimization · Large-scale optimization

1 Introduction

Post-tabular data protection methods are based on modifying or suppressing some of the table cells, yet satisfying the table additivity (that is, the sum of the "inner" cells has to be equal to the marginal cell) and preserving the original value of a subset of cells (e.g., some subtotal or total cells). This is the main

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

J. Domingo-Ferrer and M. Pejić-Bach (Eds.): PSD 2016, LNCS 9867, pp. 3–14, 2016.
DOI: 10.1007/978-3-319-45381-1_1

difference compared to pre-tabular methods, which at the same time cannot guarantee table additivity and the original value of a subset of cells. Among post-tabular data protection methods we find cell suppression [4,9] and controlled tabular adjustment [1,3], both formulating difficult mixed integer linear optimization problems. More details can be found in the monograph [12] and the survey [5].

Interval protection or partial cell suppression was introduced in [10] as a linearization of the difficult cell suppression problem. Unlike in cell suppression, interval protection replaces some cell values by intervals containing the true value. From those intervals, no attacker can be able to recompute the true value within some predefined lower and upper protection levels. One of the great advantages of interval suppression against alternative approaches is that the resulting optimization problem is convex and continuous, which means that theoretically it can be efficiently solved in polynomial time by, for instance, interior-point methods [13]. Therefore, theoretically, this approach is valid for big tables from the big-data era.

However, attempting to solve the resulting "monolithic" linear optimization model by some state-of-the-art solver is almost impossible for huge tables: we will either exhaust the RAM memory of the computer, or we will require a large CPU time. Alternative approaches to be tried include a Benders decomposition of this huge linear optimization problem. In this work we present preliminary results with a prototype implementation of Benders decomposition. A similar approach was used in the seminal publication [10] about partial cell suppression. However, this work differs in two substantial aspects: (i) our implementation considers two independent boundaries for each cell interval, whereas those two boundaries were forced to satisfy a ratio in the code of [10] (that is, actually only one boundary was considered in the 2003 code, thus solving a simpler variant of the problem); (ii) we applied our prototype to a set of seven general and hierarchical tables, where results for only three two-dimensional tables were reported in [10]. As we will see, our "not-too efficient and tuned" classical Benders decomposition prototype still outperforms state-of-the-art solvers in these complex tables.

The paper is organized as follows. Section 2 describes the general interval protection method. Section 3 outlines the Benders solution approach. The particular form of Benders for interval protection is shown in Sect. 4, which is illustrated by a small example in Subsect. 4.1. Finally, Sect. 5 reports computational results with some general and hierarchical tables.

2 The General Interval Protection Problem Formulation

We are given a table (i.e., a set of cells $a_i, i \in \mathcal{N} = \{1, \dots, n\}$), satisfying m linear relations $Aa = b$, $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. Any set of values x satisfying $Ax = b, l \le x \le u$, is a valid table, $l \in \mathbb{R}^n$, $u \in \mathbb{R}^n$ being known a priori lower and upper bounds for cell values. For positive tables we have $l_i = 0,\ u_i = +\infty,\ i = 1, \dots, n$, but the procedure here outlined is also valid for general tables.

For instance, we may consider the cells provide information about some attribute for several individual states (e.g., member states of European Union), as well as the highest-level of aggregated information (e.g., at European Union level). The set of multi-state cells, or cells providing this highest-level of aggregated information could be the ones to be replaced by intervals, and they will be denoted as $\mathcal{H} \subseteq \mathcal{N}$.

Let $\mathcal{F}, \mathcal{S}, \mathcal{M}$ be a partition of $\mathcal{N}$, i.e., $\mathcal{N} = \mathcal{F} \cup \mathcal{S} \cup \mathcal{M}$, and $\mathcal{F} \cap \mathcal{S} = \mathcal{F} \cap \mathcal{M} = \mathcal{S} \cap \mathcal{M} = \emptyset$. $\mathcal{S}$ is the set of sensitive cells to be protected, with upper and lower protection levels upl_s and lpl_s for each cell $s \in \mathcal{S}$. $\mathcal{F}$ is the set of cells whose values are known (e.g., they have been previously published by individual states). $\mathcal{M}$ is the set of non-sensitive and non previously published cells. To simplify the formulation of the forthcoming optimization problems, we can assume that for $f \in \mathcal{F}$ we have $l_f = u_f = a_f$, and then cells from $\mathcal{F}$ can be considered elements of $\mathcal{M}$, that is, $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{F}$ and $\mathcal{F} \leftarrow \emptyset$. Following our example, we have that, in general, cells in $\mathcal{S}$ provide information at state level, but in some cases multi-state cells may also be sensitive; thus we may have $\mathcal{S} \cap \mathcal{H} \neq \emptyset$. In a similar way, since multi-state cells may not have been previously published we may also have $\mathcal{M} \cap \mathcal{H} \neq \emptyset$. To make the formulation more general our only assumption will be that $\mathcal{H} \subseteq \mathcal{N}$. When $\mathcal{H} = \mathcal{N}$ we just have the standard "interval protection" or "partial cell suppression" introduced in [10].

Our purpose is to publish the set of smallest intervals $[lb_h, ub_h]$—where $l_h \leq lb_h$ and $ub_h \leq u_h$— for each cell $h \in \mathcal{H}$ instead of the real value $a_h \in [lb_h, ub_h]$, such that, from these intervals, no attacker can determine that $a_s \in (a_s - lpl_s, a_s + upl_s)$ for all sensitive cells $s \in \mathcal{S}$. This means that

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

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

$$\begin{array}{lll} \underline{a_s} = \min x_s & & \overline{a_s} = \max x_s \\ \quad \text{s.to } Ax = b & & \quad \text{s.to } Ax = b \\ \quad\quad l_i \leq x_i \leq u_i \quad i \in \mathcal{N} \setminus \mathcal{H} & \text{and} & \quad\quad l_i \leq x_i \leq u_i \quad i \in \mathcal{N} \setminus \mathcal{H} \\ \quad\quad lb_i \leq x_i \leq ub_i \ i \in \mathcal{H} & & \quad\quad lb_i \leq x_i \leq ub_i \ i \in \mathcal{H} \end{array} \tag{2}$$

Clearly, for cells $i \in \mathcal{H} \cap \mathcal{S}$, (1) and (2) imply that $lb_i \leq a_i - lpl_i$ and $ub_i \geq a_i + upl_i$.

The previous problem can be formulated as a large-scale linear optimization problem. For each primary cell $s \in \mathcal{S}$, two auxiliary vectors $x^{l,s} \in \mathbb{R}^n$ and $x^{u,s} \in \mathbb{R}^n$ are introduced to impose, respectively, the lower and upper protection requirement of (1). The problem formulation is as follows:

$$
\begin{array}{l}
\min \sum_{i \in \mathcal{H}} w_i(ub_i - lb_i) \\
\text{s.to} \\
\left.
\begin{array}{rcll}
 & Ax^{l,s} & = b & \\
l_i \leq & x_i^{l,s} & \leq u_i & i \in \mathcal{N} \setminus \mathcal{H} \\
lb_i \leq & x_i^{l,s} & \leq ub_i & i \in \mathcal{H} \\
 & x_s^{l,s} & \leq a_s - lpl_s & \\
 & & & \\
 & Ax^{u,s} & = b & \\
l_i \leq & x_i^{u,s} & \leq u_i & i \in \mathcal{N} \setminus \mathcal{H} \\
lb_i \leq & x_i^{u,s} & \leq ub_i & i \in \mathcal{H} \\
 & x_s^{u,s} & \geq a_s + upl_s &
\end{array}
\right\} \quad \forall\ s \in \mathcal{S} \\
\\
\quad l_i \leq lb_i \leq a_i \quad i \in \mathcal{H} \\
\quad a_i \leq ub_i \leq u_i \quad i \in \mathcal{H}
\end{array}
\tag{3}
$$

where w_i is a weight for the information loss associated with cell a_i.

Problem (3) is very large (easily in the order of millions of variables and constraints), but it is linear (no binary, no integer variables), and thus theoretically it can be efficiently solved in polynomial time by general or by specialized interior-point algorithms [7,13].

3 Outline of Benders Decomposition

Benders decomposition [2] was suggested for problems with two types of variables, one of them considered as "complicating variables". In MILP models complicating variables are the binary/integer ones; in continuous problems, the complicating variables are usually associated to linking variables between groups of constraints (i.e., variables lb and ub in (3)). Consider the following primal problem (P) with two groups of variables (x, y)

$$
(P) \qquad \begin{array}{ll}
\min & c^\top x + d^\top y \\
\text{s. to} & A_1 x + A_2 y = b \\
 & x \geq 0 \\
 & y \in Y,
\end{array}
$$

where y are the complicating variables, $c, x \in \mathbb{R}^{n_1}$, $d, y \in \mathbb{R}^{n_2}$, $A_1 \in \mathbb{R}^{m \times n_1}$ and $A_2 \in \mathbb{R}^{m \times n_2}$. Fixing some $y \in Y$, we obtain:

$$
(Q) \qquad \begin{array}{ll}
\min & c^\top x \\
\text{s. to} & A_1 x = b - A_2 y \\
 & x \geq 0.
\end{array}
$$

The dual of (Q) is:

$$(Q_D) \qquad \begin{array}{ll} \max & u^\top(b - A_2 y) \\ \text{s. to} & A_1^\top u \le c \\ & u \in \mathbb{R}^m. \end{array}$$

It is known that if (Q_D) has a solution then (Q) has a solution too, and both objective functions coincide; if (Q_D) is unbounded, then (Q) is infeasible. Let assume that (Q_D) is never infeasible (in the interval protection problem this is always the case). If, as notation convention, we consider that the objective of (Q) is $+\infty$ when it is infeasible, then (P) can be written as

$$(P') \qquad \begin{array}{ll} \min & \left\{ d^\top y + \max \left\{ u^\top(b - A_2 y) | A_1^\top u \le c, u \in \mathbb{R}^m \right\} \right\} \\ \text{s. to} & y \in Y. \end{array}$$

Let $U = \left\{ u | A_1^\top u \le c, u \in \mathbb{R}^m \right\}$ be the convex feasible set of (Q_D). By Minkowski representation we know that every point $u \in U$ may be represented as a convex combination of the vertices $u^1, \ldots, u^s$ and extreme rays $v^1, \ldots, v^t$ of the convex polytope U. Therefore any $u \in U$ may be written as

$$\begin{array}{c} u = \sum_{i=1}^s \lambda_i u^i + \sum_{j=1}^t \mu_j v^j \\ \sum_{i=1}^s \lambda_i = 1 \\ \lambda_i \ge 0 \quad i = 1, \ldots, s \\ \mu_j \ge 0 \quad j = 1, \ldots, t. \end{array}$$

If ${v^j}^\top(b - A_2 y) > 0$ for some $j \in \{1, \ldots, t\}$ then (Q_D) is unbounded, and thus (Q) is infeasible. We then impose

$${v^j}^\top(b - A_2 y) \le 0 \quad j = 1, \ldots, t.$$

The optimal solution of (Q_D) is then known to be in a vertex of U, and (P') may be rewritten as

$$(P'') \qquad \begin{array}{ll} \min & d^\top y + \max\limits_{i=1,\ldots,s} ({u^i}^\top(b - A_2 y)) \\ \text{s. to} & {v^j}^\top(b - A_2 y) \le 0 \quad j = 1, \ldots, t \\ & y \in Y. \end{array}$$

Introducing variable θ, (P'') is equivalent to the Benders problem (BP):

$$(BP) \qquad \begin{array}{lll} \min & \theta & \\ \text{s. to} & \theta \ge d^\top y + {u^i}^\top(b - A_2 y) & i = 1, \ldots, s \\ & {v^j}^\top(b - A_2 y) \le 0 & j = 1, \ldots, t \\ & y \in Y. & \end{array}$$

Problem (BP) is impractical since s and t can be very large, and in addition the vertices and extreme rays are unknown. Instead, the method considers a relaxation (BP_r) with a subset of the vertices and extreme rays. The relaxed Benders problem (or master problem) is thus:

$$(BP_r) \qquad \begin{array}{lll} \min & \theta & \\ \text{s. to} & \theta \geq d^\top y + {u^i}^\top (b - A_2 y) & i \in I \subseteq \{1,\ldots,s\} \\ & {v^j}^\top (b - A_2 y) \leq 0 & j \in J \subseteq \{1,\ldots,t\} \\ & y \in Y. & \end{array}$$

Initially $I = J = \emptyset$, and new vertices and extreme rays provided by the subproblem (Q_D) are added to the master problem, until the optimal solution is found. In summary, the steps of the Benders algorithm are:

Benders algorithm

0. Initially $I = \emptyset$ and $J = \emptyset$. Let (θ_r^*, y_r^*) be the solution of current master problem (BP_r), and (θ^*, y^*) the optimal solution of (BP).
1. Solve master problem (BP_r) obtaining θ_r^* and y_r^*. At first iteration, $\theta_r^* = -\infty$ and y_r is any feasible point in Y.
2. Solve subproblem (Q_D) using $y = y_r^*$. There are two cases:
   (a) (Q_D) has finite optimal solution in vertex u^{i0}.
      – If $\theta_r^* = d^\top y_r^* + {u^{i_0}}^\top (b - A_2 y_r^*)$ then **STOP**. Optimal solution is $y^* = y_r^*$ with cost $\theta^* = \theta_r^*$.
      – If $\theta_r^* < d^\top y_r^* + {u^{i_0}}^\top (b - A_2 y_r^*)$ then this solution violates constraint of (BP) $\theta > d^\top y + {u^{i_0}}^\top (b - A_2 y)$. Add this new constraint to (BP_r): $I \leftarrow I \cup \{i_0\}$.
   (b) (Q_D) is unbounded along segment $u^{i_0} + \lambda v^{j_0}$ (u^{i_0} is current vertex, v^{j_0} is extreme ray). Then this solution violates constraint of (BP) ${v^{j_0}}^\top (b - A_2 w) \leq 0$. Add this new constraint to (BP_r): $J \leftarrow J \cup \{j_0\}$; vertex may also be added: $I \leftarrow I \cup \{i_0\}$.
3. Go to step 1 above.

Convergence is guaranteed since at each iteration one or two constraints are added to (BP_r), no constraints are repeated, and the maximum number of constraints is $s + t$.

4 Benders Decomposition for the Interval Protection Problem

Problem (3) has two groups of variables: $x^{l,s} \in \mathbb{R}^n$, $x^{u,s} \in \mathbb{R}^n$; and $lb \in \mathbb{R}^{|\mathcal{H}|}$, $ub \in \mathbb{R}^{|\mathcal{H}|}$, which can be seen as the complicating variables, since if they are fixed, the resulting problem in variables $x^{l,s}$ and $x^{u,s}$ is separable, as shown below. Indeed, projecting out the $x^{l,s}$, $x^{u,s}$ variables, (3) can be written as

$$
\begin{aligned}
\min \; & \sum_{i\in\mathcal{H}} w_i(ub_i - lb_i) + Q(ub, lb) \\
\text{s.to } \; & l_i \le lb_i \le a_i \quad i \in \mathcal{H} \\
& a_i \le ub_i \le u_i \quad i \in \mathcal{H}
\end{aligned} \tag{4}
$$

where

$$
Q(ub, lb) = \min \sum_{s\in\mathcal{S}} (0_n^\top x^{l,s} + 0_n^\top x^{u,s}) = 0
$$

$$
\text{s.to} \quad \left.
\begin{array}{rlll}
 & Ax^{l,s} & = b & \\
l_i \le & x_i^{l,s} & \le u_i & i \in \mathcal{N}\setminus\mathcal{H} \\
lb_i \le & x_i^{l,s} & \le ub_i & i \in \mathcal{H} \\
 & x_s^{l,s} & \le a_s - lpl_s & \\
 & & & \\
 & Ax^{u,s} & = b & \\
l_i \le & x_i^{u,s} & \le u_i & i \in \mathcal{N}\setminus\mathcal{H} \\
lb_i \le & x_i^{u,s} & \le ub_i & i \in \mathcal{H} \\
 & x_s^{u,s} & \ge a_s + upl_s &
\end{array}
\right\} \quad \forall\, s \in \mathcal{S}, \tag{5}
$$

$0_n \in \mathbb{R}^n$ denoting the zero vector. Problem (5) is separable in the $x^{l,s}$, $x^{u,s}$ variables for each $s \in \mathcal{S}$ so it can be replaced by the solution of $2|\mathcal{S}|$ smaller problems of the form

$$
\begin{array}{rrrlll}
Q^{l,s}(ub, lb) = & \min \; 0_n^\top x^{l,s} = 0 & & & & \\
& \text{s.to} & & Ax^{l,s} & = b & \\
& & l_i \le & x_i^{l,s} & \le u_i & i \in \mathcal{N}\setminus\mathcal{H} \\
& & lb_i \le & x_i^{l,s} & \le ub_i & i \in \mathcal{H} \\
& & & x_s^{l,s} & \le a_s - lpl_s \;, &
\end{array} \tag{6}
$$

for the lower protection of sensitive cell $s \in \mathcal{S}$, and

$$
\begin{array}{rrrlll}
Q^{u,s}(ub, lb) = & \min \; 0_n^\top x^{u,s} = 0 & & & & \\
& \text{s.to} & & Ax^{u,s} & = b & \\
& & l_i \le & x_i^{u,s} & \le u_i & i \in \mathcal{N}\setminus\mathcal{H} \\
& & lb_i \le & x_i^{u,s} & \le ub_i & i \in \mathcal{H} \\
& & & x_s^{u,s} & \ge a_s + upl_s \;. &
\end{array} \tag{7}
$$

for the upper protection of sensitive cell $s \in \mathcal{S}$. Note that (5)–(7) are just feasibility problems with a constant (dummy) objective function.

Problems (6) and (7) are our Benders subproblems. Due to its constant objective function, (6) and (7) are feasibility problems. Therefore Benders algorithm will only include extreme rays of the dual formulations of (6) and (7) to guarantee the feasibility of the values of lb and ub provided by the master problem.

Denoting the j-th extreme ray of the dual formulation of (6) as $v_{l,s}^j = (v_j^{\lambda^{l,s}}, v_j^{\mu_l^{l,s}}, v_j^{\mu_u^{l,s}}, v_j^{\nu^{l,s}})$, where $\lambda^{l,s}$, $\mu_l^{l,s}$, $\mu_u^{l,s}$ and $\nu^{l,s}$ refer to the indices of the

Lagrange multipliers of the constraints of (6), it can be shown that the feasibility cut to be added to the master problem would be

$$0 \geq \sum_{i=1}^{m} v_{j,i}^{\lambda^{l,s}} b_i + \sum_{i \in \mathcal{N}\setminus\mathcal{H}} (-v_{j,i}^{\mu_u^{l,s}} u_i + v_{j,i}^{\mu_l^{l,s}} l_i) + \sum_{i \in \mathcal{H}} (-v_{j,i}^{\mu_u^{l,s}} ub_i + v_{j,i}^{\mu_l^{l,s}} lb_i) - (a_s - lpl_s) v_j^{\nu^{l,s}} \\ = g_{l,s}^{j}(ub, lb). \tag{8}$$

The extreme rays of the dual of (7) have an analogous form $v_{u,s}^j = (v_j^{\lambda^{u,s}}, v_j^{\mu_l^{u,s}}, v_j^{\mu_u^{u,s}}, v_j^{\nu^{u,s}})$ and so does the feasibility cut to be added to the master problem:

$$0 \geq \sum_{i=1}^{m} v_{j,i}^{\lambda^{u,s}} b_i + \sum_{i \in \mathcal{N}\setminus\mathcal{H}} (-v_{j,i}^{\mu_u^{u,s}} u_i + v_{j,i}^{\mu_l^{u,s}} l_i) + \sum_{i \in \mathcal{H}} (-v_{j,i}^{\mu_u^{u,s}} ub_i + v_{j,i}^{\mu_l^{u,s}} lb_i) + (a_s - lpl_s) v_j^{\nu^{u,s}} \\ = g_{u,s}^{j}(ub, lb). \tag{9}$$

Denoting as $\mathcal{I}_{l,s}$ and $\mathcal{I}_{u,s}$ the set of indices of feasibility cuts obtained from $Q^{l,s}$ and $Q^{u,s}$, the master problem is:

$$\begin{array}{lll} \min & \sum_{i \in \mathcal{H}} w_i (ub_i - lb_i) & \\ \text{s.to} & g_j^{l,s}(ub, lb) \leq 0 & j \in \mathcal{I}_{l,s} \\ & g_j^{u,s}(ub, lb) \leq 0 & j \in \mathcal{I}_{u,s} \\ & l_i \leq lb_i \leq a_i & i \in \mathcal{H} \\ & a_i \leq ub_i \leq u_i & i \in \mathcal{H}. \end{array} \tag{10}$$

The Benders decomposition algorithm will then solve (10) for the master problem and the duals of (6) and (7) for the subproblems.

4.1 Illustrative Example

Consider the following simple table

10	15	25
20	17	37

of $n = 6$ cells and $m = 2$ linear constraints associated to row totals

$$\begin{array}{l} a_1 + a_2 - a_3 = 0 \\ a_4 + a_5 - a_6 = 0 \end{array}$$

(we don't consider column totals to simplify the example), where $\mathcal{H} = \mathcal{N} = \{1, \dots, 6\}$, and a_1 and a_5 as the two sensitive cells, whose parameters are given by

s	a_s	lpl_s	upl_s
1	10	5	5
5	17	7	4

.

Note that this example, in principle, can not be solved with the original implementation of [10] since the ratios between upper and lower protection levels are not the same for all sensitive cells.

We next show the application of Benders algorithm to the previous table:

1. **Initialization.**
The number of cuts for the lb and the ub variables is set to 0, this means $\mathcal{I}_{l,s} = \mathcal{I}_{u,s} = \emptyset$. The first master problem to be solved is thus

$$\begin{array}{ll} \min \sum_{i=1}^{6}(ub_i - lb_i) & \\ \text{s.to } l_i \leq lb_i \leq a_i & i = 1, \ldots, 6 \\ \quad a_i \leq ub_i \leq u_i & i = 1, \ldots, 6, \end{array}$$

obtaining some initial values for lb, ub.
2. **Iterating Through Benders' Algorithm.**

Cut generation is based on (8)–(9), details are omitted to simplify the exposition.
- **Iteration 1.** The two Benders cuts obtained for cell 1 are $lb_1 \leq 5$ and $ub_1 \geq 15$. The two Benders cuts obtained for cell 5 are $lb_5 \leq 10$ and $ub_1 \geq 21$. Note these are obvious cuts associated to the protection levels of sensitive cells, that could have been added from the beginning in an efficient implementation, thus avoiding this first Benders iteration.
- **Iteration 2.** The current master subproblem

$$\begin{array}{ll} \min \sum_{i=1}^{6}(ub_i - lb_i) & \\ \text{s.to } l_i \leq lb_i \leq a_i & i \in 1, \ldots, 6 \\ \quad a_i \leq ub_i \leq u_i & i \in 1, \ldots, 6 \\ \quad lb_1 \leq 5, & ub_1 \geq 15 \\ \quad lb_5 \leq 10, & ub_5 \geq 21 \end{array}$$

has solution $lb = [5, 15, 25, 20, 10, 37]$ and $ub = [15, 15, 25, 20, 21, 37]$. Using this solution the two Benders cuts obtained for cell 1 are $lb_3 - ub_2 \leq 58$ and $lb_2 - ub_2 \geq 15$. The two cuts obtained for cell 5 are $lb_6 - ub_4 \leq 21$ and $ub_6 - lb_4 \geq 39$.
- **Iteration 3.** The current master problem is

$$\begin{array}{ll} \min \sum_{i=1}^{6}(ub_i - lb_i) & \\ \text{s.to } l_i \leq lb_i \leq a_i & i \in 1, \ldots, 6 \\ \quad a_i \leq ub_i \leq u_i & i \in 1, \ldots, 6 \\ \quad lb_1 \leq 5, & ub_1 \geq 15 \\ \quad lb_5 \leq 10, & ub_5 \geq 21 \\ \quad lb_3 - ub_2 \leq 58, & lb_2 - ub_2 \geq 15 \\ \quad lb_6 - ub_4 \leq 21, & ub_6 - lb_4 \geq 39, \end{array}$$

with solution $lb = [5, 15, 20, 16, 10, 30]$ and $ub = [15, 15, 30, 20, 21, 37]$. Benders subproblems happen to be feasible with these values, thus we have an optimal solution of objective $\sum_{i=1}^{6}(ub_i - lb_i) = 42$. Since this table is small, the original model was solved using some off-the-shelf optimization solver, obtaining the same optimal objective function.

3. **Auditing.** Although this step is not needed with interval protection, to be sure that this solution satisfies that no attacker can determine that $a_s \in (a_s - lpl_s, a_s + upl_s)$ for $s \in \{1, 5\}$, the problems (2) were solved, obtaining $\underline{a_1} = 5$, $\overline{a_1} = 15$, $\underline{a_5} = 10$ and $\overline{a_5} = 21$. Therefore, it can be asserted that it is safe to publish this solution.
4. **Publication of the table.** The final safe table to be published would be

[5, 15]	15	[20, 30]
[16, 20]	[10, 21]	[30, 37]

.

5 Computational Results

We developed a prototype implementation of the Benders algorithm for interval protection using the AMPL modeling language [11] and Cplex for the master and subproblems. We solved seven instances, whose dimensions are given in Table 1. Columns n, $|\mathcal{S}|$ and m provide, respectively, the number of cells, sensitive cells and table linear equations. Table "targus" is a general table, while the remaining six tables are 1H2D tables (i.e., two-dimensional hierarchical tables with one hierarchical variable) obtained with a generator used in the literature [1,8].

Table 1. Instance dimensions and results with Benders decomposition

Table	n	$\|\mathcal{S}\|$	m	CPU	it$_B$	it$_S$	obj
Targus	162	13	63	5.17	31	8872	2142265.7
Table 1	121	10	55	3.41	26	7167	136924
Table 2	1680	158	299	410.53	43	1104884	43715149
Table 3	600	53	170	26.38	43	131834	3624906
Table 4	756	68	243	50.92	33	144963	9134139
Table 5	168	14	62	3.95	19	5959	303844
Table 6	1584	143	485	966.28	70	1729767	21302104

The results obtained with the Benders decomposition are provided in the last columns of Table 1. Columns "CPU", "it$_B$", "it$_S$" and "obj" provide respectively the total CPU time, number of Benders iterations, overall number of simplex iterations, and the final optimal objective function obtained.

Table 2 provides results for the solution of the monolithic model (3) using Cplex default linear algorithm (dual simplex). Column "n.var" reports the number of variables of the resulting linear optimization problem. The meaning of remaining columns is the same as in Table 1. Three executions, clearly marked, were aborted because the CPU time was excessive compared with the solution

Table 2. Results using Cplex for monolithic model

Table	CPU	it_S	n.var	obj
Targus	36.0515	16532	4212	2142265.7
Table 1	3.43548	7452	2420	136924
Table 2	2944.87[a]	—	530880	16056608400
Table 3	522.875[a]	—	63600	260592812
Table 4	11085.6	436895	102816	9134139
Table 5	10.6764	17325	4704	303844
Table 6	7816.61[a]	—	453024	4404161015

[a] Aborted due to excessive CPU time

by Benders; in those cases column "obj" provides the value of the objective function when the algorithm was stopped. From these tables it is clear that the solution of the monolithic model is impractical and that an standard implementation of Benders can be more efficient for some classes of problems (namely, 1H2D tables).

6 Conclusions

Partial cell suppression or interval protection can be an alternative method for tabular data protection. Unlike other approaches, this method results in a huge but continuous optimization problem, which can be effectively solved by linear optimization algorithms. One of them is Benders decomposition: a prototype code was able to solve some nontrivial tables more efficiently than state-of-the-art solvers applied to the monolithic model. It is expected that a more sophisticated implementation of Benders algorithm would be able to solve even larger and more complex tables. An additional and promising line of research would be to consider highly efficient specialized interior-point methods for block-angular problems [6,7]. This is part of the further work to be done.

References

1. Baena, D., Castro, J., González, J.A.: Fix-and-relax approaches for controlled tabular adjustment. Comput. Oper. Res. **58**, 41–52 (2015)
2. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Comput. Manag. Sci. **2**, 3–19 (2005). English translation of the original paper appeared in Numerische Mathematik **4**, 238–252 (1962)
3. Castro, J.: Minimum-distance controlled perturbation methods for large-scale tabular data protection. Eur. J. Oper. Res. **171**, 39–52 (2006)
4. Castro, J.: A shortest paths heuristic for statistical disclosure control in positive tables. INFORMS J. Comput. **19**, 520–533 (2007)
5. Castro, J.: Recent advances in optimization techniques for statistical tabular data protection. Eur. J. Oper. Res. **216**, 257–269 (2012)

6. Castro, J.: Interior-point solver for convex separable block-angular problems. Optim. Methods Softw. **31**, 88–109 (2016)
7. Castro, J., Cuesta, J.: Quadratic regularizations in an interior-point method for primal block-angular problems. Math. Program. **130**, 415–445 (2011)
8. Castro, J., Frangioni, A., Gentile, C.: Perspective reformulations of the CTA problem with L_2 distances. Oper. Res. **62**, 891–909 (2014)
9. Fischetti, M., Salazar, J.J.: Solving the cell suppression problem on tabular data with linear constraints. Manag. Sci. **47**, 1008–1026 (2001)
10. Fischetti, M., Salazar, J.J.: Partial cell suppression: a new methodology for statistical disclosure control. Stat. Comput. **13**, 13–21 (2003)
11. Fourer, R., Gay, D.M., Kernighan, D.W.: AMPL: A Modeling Language for Mathematical Programming, 2nd edn. Thomson Brooks/Cole, Pacific Grove (2003)
12. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Schulte-Nordholt, E., Spicer, K., de Wolf, P.P.: Statistical Disclosure Control. Wiley, Chichester (2012)
13. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM, Philadelphia (1997)