Using a Mathematical Programming Modeling Language for Optimal CTA

Jordi Castro[1,⋆,⋆⋆] and Daniel Baena[2]

[1] Department of Statistics and Operations Research,
Universitat Politècnica de Catalunya,
Jordi Girona 1–3, 08034 Barcelona, Catalonia
jordi.castro@upc.edu
http://www-eio.upc.es/~jcastro
[2] Institut d'Estadística de Catalunya,
Via Laietana 58, 08003 Barcelona, Catalonia
dbaena@idescat.net

Abstract. Minimum-distance controlled tabular adjustment methods (CTA) have been formulated as an alternative to the cell suppression problem (CSP) for tabular data. CTA formulates an optimization problem with fewer variables and constraints than CSP. However, the inclusion of binary decisions about protection sense of sensitive cells (optimal CTA) in the formulation, still results in a mixed integer-linear problem. This work shows how mathematical programming modeling languages can be used to develop a prototype for optimal CTA based on Benders method. Preliminary results are reported for some medium size two-dimensional tables. For this type of tables, the approach is competitive with other general-purpose algorithms implemented in commercial solvers.

Keywords: statistical disclosure control, controlled tabular adjustment, mixed-integer linear programming, Benders decomposition.

1 Introduction

Minimum-distance controlled tabular adjustment methods (CTA) were suggested in [2,7] as an alternative to the difficult cell suppression problem (CSP) [3,8]. In some instances, the quality of CTA solutions has shown to be higher than that provided by CSP ones [4].

Although CTA formulates an optimization problem with fewer variables and constraints than CSP, it is also a mixed integer-linear problem (MILP) if the binary decision of protection sense of sensitive cells is included in the model (optimal CTA). Therefore, for some instances, the solution time of optimal CTA by a general purpose solver, like CPLEX or XPress, can still be large. (Some

⋆ Supported by grant MTM2006-05550 of the Spanish Ministry of Science and Education.

⋆⋆ Corresponding author.

J. Domingo-Ferrer and Y. Saygın (Eds.): PSD 2008, LNCS 5262, pp. 1–12, 2008.

metaheuristics approaches have been used, but only for small-medium instances [6].) For MILP models there are some specialized algorithms. One of them is Benders method [1]. In this work we show how a mathematical programming modeling language can be used for a prototype for optimal CTA based on Benders decomposition. Preliminary results with this prototype are reported, using a battery of two-dimensional tables. For these instances, the algorithm is more efficient that the general purpose solver implemented in CPLEX.

The paper is organized as follows. Section 2 reviews the CTA method. Section 3 outlines the Benders decomposition algorithm for the non mathematical programming experts. Section 4 shows how this approach can be implemented in the AMPL mathematical programming language. Section 5 illustrates the approach in the solution of a small example. Finally, Section 6 reports computational results in the solution of a set of two-dimensional tables.

2 The Optimal CTA Problem

Given (i) a set of cells $a_i, i = 1, \ldots, n$, that satisfy some linear relations $Aa = b$ (a being the vector of a_i's); (ii) a lower and upper bound for each cell $i = 1, \ldots, n$, respectively l_{a_i} and u_{a_i}, which are considered to be known by any attacker; (iii) a set $\mathcal{P} = \{i_1, i_2, \ldots, i_p\} \subseteq \{1, \ldots, n\}$ of indices of sensitive cells; (iv) and a lower and upper protection level for each sensitive cell $i \in \mathcal{P}$, respectively lpl_i and upl_i, such that the released values satisfy either $x_i \geq a_i + upl_i$ or $x_i \leq a_i - lpl_i$; the purpose of CTA is to find the closest safe values $x_i, i = 1, \ldots, n$, according to some distance L, that makes the released table safe. This involves the solution of the following optimization problem:

$$
\begin{array}{ll}
\min\limits_{x} & ||x - a||_L \\
\text{s. to} & Ax = b \\
& l_{a_i} \leq x_i \leq u_{a_i} \quad i = 1, \ldots, n \\
& x_i \leq a_i - lpl_i \text{ or } x_i \geq a_i + upl_i \quad i \in \mathcal{P}.
\end{array}
\tag{1}
$$

Problem (1) can also be formulated in terms of deviations from the current cell values. Defining $z_i = x_i - a_i, \quad i = 1, \ldots, n$ —and similarly $l_{z_i} = l_{x_i} - a_i$ and $u_{z_i} = u_{x_i} - a_i$—, (1) can be recast as:

$$
\begin{array}{ll}
\min\limits_{z} & ||z||_L \\
\text{s. to} & Az = 0 \\
& l_{z_i} \leq z_i \leq u_{z_i} \quad i = 1, \ldots, n \\
& z_i \leq -lpl_i \text{ or } z_i \geq upl_i \quad i \in \mathcal{P},
\end{array}
\tag{2}
$$

$z \in \mathbb{R}^n$ being the vector of deviations. Using the L_1 distance, and after some manipulation, (2) can be written as

$$
\begin{array}{lll}
\min\limits_{z^+,z^-,y} & \sum\limits_{i=1}^{n} w_i(z_i^+ + z_i^-) & \\
\text{s. to} & A(z^+ - z^-) = 0 & \\
& 0 \le z_i^+ \le \quad u_{z_i} \quad i \notin \mathcal{P} & \\
& 0 \le z_i^- \le -l_{z_i} \quad i \notin \mathcal{P} & \\
& upl_i \; y_i \le z_i^+ \le u_{zi} \; y_i \quad i \in \mathcal{P} & \\
& lpl_i(1-y_i) \le z_i^- \le -l_{zi}(1-y_i) \quad i \in \mathcal{P}, &
\end{array} \qquad (3)
$$

$w \in \mathbb{R}^n$ being the vector of cell weights, $z^+ \in \mathbb{R}^n$ and $z^- \in \mathbb{R}^n$ the vector of positive and negative deviations in absolute value, and $y \in \mathbb{R}^p$ being the vector of binary variables associated to protections senses. When $y_i = 1$ the constraints mean $upl_i \le z_i^+ \le u_{zi}$ and $z_i^- = 0$, thus the protection sense is “upper”; when $y_i = 0$ we get $z_i^+ = 0$ and $lpl_i \le z_i^- \le -l_{z_i}$, thus protection sense is “lower”. Model (3) is a (difficult) MILP.

3 Outline of Benders Method for MILP Problems

Benders decomposition method [1] 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. Consider the following MILP primal problem (P) in variables (x, y)

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

where y are the binary/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 $A_2 \in \mathbb{R}^{m \times n_2}$. For binary problems, as in optimal CTA, we have $Y = \{0,1\}^{n_2}$. Fixing some $y \in Y$, we obtain:

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

The dual of (Q) is:

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

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 (indeed, this is the case in optimal CTA). If, as notation convention, we consider that the objective of (Q) is $+\infty$ when it is infeasible, then (P) can be written as

$$
(P') \qquad \begin{array}{ll}
\min & \left\{ d^T y + \max\left\{ u^T(b - A_2 y) | A_1^T u \le c, u \in \mathbb{R}^m \right\} \right\} \\
\text{s. to} & y \in Y.
\end{array}
$$

Let $U = \left\{u | A_1^T u \leq c, u \in \mathbb{R}^m\right\}$ 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

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

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

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

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

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

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

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

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:

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

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 w_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 $\theta_r^* = d^T y_r^* + u^{i_0 T}(b - A_2 y_r^*)$ then **STOP**. Optimal solution is $y^* = y_r^*$ with cost $\theta^* = \theta_r^*$.
   – If $\theta_r^* < d^T y_r^* + u^{i_0 T}(b - A_2 y_r^*)$ then this solution violates constraint of (BP) $\theta > d^T y + u^{i_0 T}(b - A_2 y)$. 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) $v^{j_0 T}(b - A_2 w) \le 0$. 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 Prototype of Benders Method for Optimal CTA

It can be shown that, applying Benders method to the optimal CTA problem (3), the formulation subproblem (Q_D) is given by (see [5] for details):

$$\begin{array}{ll} \max\limits_{\mu_u^+,\mu_u^-,\mu_l^+,\mu_l^-} & -{\mu_u^+}^T u^+ - {\mu_u^-}^T u^- + {\mu_l^+}^T l^+ + {\mu_l^-}^T l^- \\ \text{s. to} & \begin{pmatrix} A^T \\ -A^T \end{pmatrix} \lambda - \begin{pmatrix} \mu_u^+ \\ \mu_u^- \end{pmatrix} + \begin{pmatrix} \mu_l^+ \\ \mu_l^- \end{pmatrix} = \begin{pmatrix} w \\ w \end{pmatrix} \\ & \mu_u^+, \mu_u^-, \mu_l^+, \mu_l^- \ge 0, \lambda \text{ free }, \end{array} \tag{4}$$

where $\mu_u^+, \mu_u^-, \mu_l^+, \mu_l^- \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$, and l^+, l^-, u^+, u^- provide the lower and upper bounds of z^+ and z^- once binary variables $y \in \mathbb{R}^p$ are fixed.

Similarly, the formulation of the master (BP_r) is

$$\begin{array}{ll} \min\limits_{\theta,y} & \theta \\ \text{s. to} & \sum\limits_{h \notin \mathcal{P}} (-\mu_{u_h}^{+,i} u_{z_h} + \mu_{u_h}^{-,i} l_{z_h}) + \sum\limits_{h \in \mathcal{P}} (\mu_{u_h}^{-,i} l_{z_h} + \mu_{l_h}^{-,i} lpl_h) + \\ & + \sum\limits_{h \in \mathcal{P}} (-\mu_{u_h}^{+,i} u_{z_h} - \mu_{u_h}^{-,i} l_{z_h} + \mu_{l_h}^{+,i} upl_h - \mu_{l_h}^{-,i} lpl_h) y_h \le \theta \quad i \in I \\ & \sum\limits_{h \notin \mathcal{P}} (-v_{u_h}^{+,j} u_{z_h} + v_{u_h}^{-,j} l_{z_h}) + \sum\limits_{h \in \mathcal{P}} (v_{u_h}^{-,j} l_{z_h} + v_{l_h}^{-,j} lpl_h) + \\ & + \sum\limits_{h \in \mathcal{P}} (-v_{u_h}^{+,j} u_{z_h} - v_{u_h}^{-,j} l_{z_h} + v_{l_h}^{+,j} upl_h - v_{l_h}^{-,j} lpl_h) y_h \le 0 \quad j \in J \\ & y_h \in \{0,1\} \quad h \in \mathcal{P}. \end{array} \tag{5}$$

Constraint

$$\theta \ge \sum_{h \in \mathcal{P}} \min(lpl_h, upl_h) w_h \tag{6}$$

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

Fig. 1. Small table for optimal CTA by Benders method. Sensitive cells are in boldface. Symmetric protection limits lpl_i and upl_i are in brackets. Weights are cell values ($w_i = a_i$).

may also be imposed to (5) since the master provides a lower bound on the optimal objective, and the right-hand-side of (6) provides a known lower bound on θ. Problems (4) and (5), together with the Benders algorithm were implemented in the AMPL modeling language [9]. Appendix A shows an extract of this implementation for (4) and (5).

5 Illustrative Example

Benders method is applied to the small two-dimensional table of Figure 1:

- Initialization (Step 0): $I = J = \emptyset$, lower bound (6) is 165.
- Iteration 1
  - Step 1: Solve master problem (5) with only constraint (6), obtaining $\theta_r^* = 165$, and $y_{r_i}^* = 1$ for all $i = 0, \ldots, 4$.
  - Step 2: Solve subproblem (4), with optimal objective $458 \geq 165 = \theta_r^*$. Add first optimality cut

    $$60y_1 + 298160000y_2 + 350776000y_3 + 70155300y_4 + \theta \geq 458$$

    to (5).
- Iteration 2:
  - Step 1: Solve master problem (5):

    $$\begin{array}{ll} \min & \theta \\ \text{s. to} & 60y_1 + 298160000y_2 + 350776000y_3 + 70155300y_4 + \theta \geq 458 \\ & \theta \geq 165. \end{array}$$

    obtaining $\theta_r = 165$, $y_i = 1$ for all $i = 1, \ldots, 4$.
  - Step 2: Solve subproblem (4), with optimal objective $458 \geq 165 = \theta_r^*$. Add second optimality cut

    $$-60y_1 - 368y_2 - 304y_3 - 202y_4 + \theta \geq -476$$

    to (5).

⋮

Table 1. Summary of illustrative example

iter.	$f^*_{Q_D}$	θ^*_r	y_1	y_2	y_3	y_4
1	458	165	0	0	0	0
2	458	165	1	1	1	1
3	330	165	1	0	0	1
4	334	165	1	1	0	1
5	346	165	0	1	0	0
6	303	165	0	1	0	1
7	334	238	0	0	1	0
8	303	274	1	0	1	0
9	303	303	1	0	1	0

Table 2. Instance dimensions

Instance	n	p	m	n. coef.
dale	16514	4923	405	33028
osorio	10201	7	202	20402
table8	1271	3	72	2542
targus	162	13	63	360
random1	22801	15000	302	45602
random2	30351	12000	352	60702
random3	40401	10000	402	80802
random4	40401	20000	402	80802
random5	35376	10000	377	70752
random6	10201	6000	202	20402
random7	10201	7000	202	20402
random8	20301	15000	302	40602
random9	20301	10000	302	40602
random10	40401	30000	402	80802
random11	30351	25000	352	60702
random12	10251	8500	252	20502
random13	37901	20000	402	75802
random14	22801	20000	302	45602
random15	25351	10000	352	50702
random16	22801	10000	302	45602
random17	22801	18500	302	45602
random18	15251	13000	252	30502
random19	15251	11000	252	30502
random20	22801	18500	302	45602

– Iteration 9:
  - Step 1: Solve master problem (5):

$$
\begin{array}{ll}
\min & \theta \\
\text{s.to} & 60y_1{+}298160000y_2{+}350776000y_3{+}70155300y_4 + \theta \geq 458 \\
& -60y_1 - 368y_2 - 304y_3 - 202y_4 + \theta \geq -476 \\
& 36y_1 + 298160000y_2 + 44y_3 - 90y_4 + \theta \geq 276 \\
& -320y_1 - 368y_2 - 44y_3 + 30y_4 + \theta \geq -324 \\
& -36y_1 - 72y_2 + 36y_3 + 280621000y_4 + \theta >= 274 \\
& 54y_1 + 24y_2 - 44y_3 - 418y_4 + \theta >= -91 \\
& 350776000y_1 + 298160000y_2 + 44y_3 - 30y_4 + \theta \geq 378 \\
& -54y_1 - 24y_2 + 44y_3 + 280621000y_4 + \theta \geq 293 \\
& \theta \geq 165
\end{array}
$$

    obtaining $\theta_r = 303$, and $y_1 = y_3 = 1$ and $y_2 = y_4 = 0$.
  - Step 2: Solve subproblem (4), with optimal objective $303 = 303 = \theta_r$. Solution found: $y^* = y_r^*$.

Table 1 summarizes the example, showing for each iteration the optimal objective function of the subproblem "$f^*_{Q_D}$" and the master problem "θ^*_r", and the values of y^*_r.

Table 3. Results with Benders method

Instance	CPU	iter.	MIP iter.	Simp. iter.	f_{Q_D}	θ
dale	8.47	20	2591	2783	3581.03	3549.53
osorio	73.94	123	13890	50903	6.0317	6.0073
table8	0.24	9	60	43	3.0848	3.0848
targus	0.62	16	449	399	59.3295	58.8393
random1	2.38	4	342	494	48477.7	47993
random2	3.48	5	463	775	38726.3	38398.8
random3	4.9	7	596	510	32170	31907.4
random4	4.22	4	338	466	64127.5	63522.7
random5	4.1	6	577	991	32159.7	31868
random6	1.73	7	1554	1395	12963.4	12835.9
random7	1.68	7	574	722	30250.1	29979.5
random8	3.46	6	461	661	48469	48088.3
random9	3.00	7	590	779	8852.87	8769.04
random10	5.90	5	372	642	64720.7	64111.1
random11	3.15	4	218	397	107088	106025
random12	4.07	12	1261	1282	18388	18210
random13	4.84	5	362	131	128188	127164
random14	3.15	5	337	495	170645	169344
random15	3.67	7	586	932	85189.8	84441.4
random16	4.26	9	985	1246	32339.3	32028
random17	3.24	5	468	635	59720.9	59170.5
random18	1.69	4	353	411	42052.8	41658.4
random19	2.02	5	462	553	35507.2	35209.6
random20	3.18	5	468	635	59720.9	59170.5

Table 4. Results with CPLEX

Instance	CPU	MIP iter.	f^*
dale	3.53	11559	3562.11
osorio	93.09	2093	6.0316
table8	1.08	147	3.0848
targus	0.13	107	59.3295
random1	14.26	33717	48074.07
random2	11.38	27526	38474.2
random3	8.75	23210	31998.93
random4	23.83	45053	63616.5
random5	9.41	23302	31953.81
random6	3.71	13754	12872.3
random7	4.69	16004	30063.69
random8	15.33	33749	48161.70
random9	8.2	22849	8791.66
random10	36.41	66306	64170.65
random11	26.8	55080	106138.14
random12	5.92	19329	18262.87
random13	22.42	44789	127343.58
random14	19.86	44124	169538.78
random15	9.06	23186	84688.72
random16	8.29	23228	32096.94
random17	19.08	41301	59240.23
random18	9.16	28959	41731.95
random19	8.53	24856	35280.94
random20	17.33	41301	59240.23

6 Computational Results

Benders algorithm for optimal CTA has been implemented using the AMPL mathematical programming modeling language [9]. This implementation has been applied to a set of four small pseudo-real and 20 random larger two-dimensional instances obtained with the generator used in [3]. All runs were carried on a Sun Fire V20Z server with two AMD Opteron processors (without exploiting parallelism capabilities), 8 GB of RAM, and under the Linux operating system. Table 2 show the instance dimensions: number of cells n, number of sensitive cells p (which is the number of binary variables), number of linear relations m, and number of coefficients in linear relations “n. coef.”.

Table 3 shows the results obtained with AMPL implementation of Benders method. Column “CPU” provides the CPU time for solution of the master and subproblems using CPLEX. Column “Benders iter.” gives the total number of Benders iterations. Columns “MIP iter.”and “Simp. iter.” show the total number of MIP and simplex iterations for, respectively, all masters and subproblems. Columns f_{Q_D} and θ show, respectively, the upper and lower bounds found. An optimality tolerance of 1% was used for all runs.

Table 4 shows the results with the CPLEX branch-and-cut algorithm. Column "CPU" provides the CPU time. Column "MIP iter." gives the overall number of MIP simplex iterations. Column f^* provides the optimal objective function found. As for Benders, an optimality tolerance of 1% was used for all runs. It can be observed that in all instances, but for "dale" and "targus", Benders method is faster than CPLEX. In particular, efficiency of Benders increases with the number of sensitive cells (i.e., binary variables), as in instances "random10", "random11", "random13", "random14", "random17" and "random18". This makes it a promising approach for large tables.

7 Conclusions

This work presented an AMPL implementation of Benders decomposition for optimal CTA. The main benefit of this prototype code is to have a tool for ease testing with alternative cuts. Preliminary results for some small-medium two-dimensional tables show it can be a promising approach for more complex tables, if Benders can be appropriately tuned to efficiently deal with them. The development of a more efficient code, and applying it to larger two-dimensional tables, and more complex structures, is part of the further work to be done.

References

1. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Computational Management Science 2, 3–19 (2005); English translation of the original paper appeared in Numerische Mathematik 4, 238–252 (1962)
2. Castro, J.: Minimum-distance controlled perturbation methods for large-scale tabular data protection. European Journal of Operational Research 171, 39–52 (2006)
3. Castro, J.: A shortest paths heuristic for statistical disclosure control in positive tables. INFORMS Journal on Computing 19, 520–533 (2007)
4. Castro, J., Giessing, S.: Testing variants of minimum distance controlled tabular adjustment. In: Monographs of Official Statistics, Eurostat-Office for Official Publications of the European Communities, Luxembourg, pp. 333–343 (2006)
5. Castro, J., González, J.A.: A Benders decomposition approach to CTA, working paper, Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya (2008)
6. Cox, L.H.: Confidentiality protection by CTA using metaheuristic methods. In: Monographs of Official Statistics, Eurostat-Office for Official Publications of the European Communities, Luxembourg, pp. 267–276 (2006)
7. Dandekar, R.A., Cox, L.H.: Synthetic tabular Data: an alternative to complementary cell suppression, manuscript, Energy Information Administration, U.S, Available from the first author on request (Ramesh.Dandekar@eia.doe.gov) (2002)
8. Fischetti, M., Salazar, J.J.: Solving the cell suppression problem on tabular data with linear constraints. Management Science 47, 1008–1026 (2001)
9. Fourer, R., Gay, D.M., Kernighan, D.W.: AMPL: A Modeling Language for Mathematical Programming. Duxbury Press (2002)

A AMPL Models for Benders Subproblem and Master

A.1 Extract of AMPL Implementation of (4)

```
###############################################
# Definiton of Benders subproblem for CTA
###############################################
param lp{1..ncells} default 0;
param up{i in 1..ncells} default (ub[i]-a[i]);;
param ln{1..ncells} default 0;
param un{i in 1..ncells}default (a[i]-lb[i]);

var la_up {1..ncells} >= 0;
var la_un {1..ncells} >= 0;
var la_lp{1..ncells} >= 0;
var la_ln {1..ncells} >= 0;
var lambda {1..nconstraints};

maximize QD:
    sum {i in 1..ncells} (-la_up[i]*up[i] -la_un[i]*un[i]
    + la_lp[i]*lp[i] + la_ln[i]*ln[i]);
subj to R1_QD {i in 1..ncells}:
    sum{l in Trasbegconst[i]..Trasbegconst[i+1]-1}
        Trascoef[l]*lambda[Trasxcoef[l]]
        -la_up[i] + la_lp[i] =c[i];
subj to R2_QD {i in 1..ncells}:
    sum{l in Trasbegconst[i]..Trasbegconst[i+1]-1}
        -Trascoef[l]*lambda[Trasxcoef[l]]
        -la_un[i]+la_ln[i] =c[i];
```

A.2 Extract of AMPL Implementation of (5)

```
##################################################
# Definition of Benders master for CTA
##################################################
param nCUT >= 0 integer;
param iter >= 0 integer;
param mipgap;
param const {1..nCUT} default 0;
param consty {1..npcells,1..nCUT} default 0;
param cut_type {1..nCUT} symbolic within {"point","ray"};
param MinTheta;
```

```
var y {1..npcells} binary;
var Theta;

minimize BPr: Theta;
#Feasibility/optimality cuts
subj to Cut_Point {j in 1..nCUT}:
    if (cut_type[j]="point") then Theta else 0 >=
    const[j] + sum {i in 1..npcells} consty[i,j]*y[i];
subj to RMinTheta:  Theta >= MinTheta;
```