A Computational Evaluation of Optimization Solvers for CTA[⋆]

Jordi Castro

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.edu/~jcastro

Abstract. Minimum-distance controlled tabular adjustment methods (CTA), an its variants, are considered an emerging perturbative approach for tabular data protection. Given a table to be protected, the purpose of CTA is to find the closest table that guarantees protection levels for the sensitive cells. We consider the most general CTA formulation which includes binary variables, thus providing protected tables with a higher data utility, at the expense of a larger solution time. The resulting model is a Mixed Integer Linear Problem (MILP). The purpose of this work is twofold. First, it presents and describes the main features of a package for CTA which is linked to both commercial (Cplex and Xpress) and open-source (Glpk, Cbc and Symphony) MILP solvers. The particular design of the package allows easy integration with additional solvers. The second objective is to perform a computational evaluation of the above two commercial and three open-source MILP solvers for CTA, using both standard instances in the literature and real-world ones. Users of tabular data confidentiality techniques in National Statistical Agencies may find this information useful for the trade-off between the (more efficient but expensive) commercial and the (slower but free) open-source MILP solvers.

Keywords: statistical disclosure control, controlled tabular adjustment, mixed integer linear programming, optimization software, open-source software.

1 Introduction

According to the recent handbook [12], minimum-distance controlled tabular adjustment methods (CTA) are considered an emerging technology for the protection of tabular data. CTA was initially suggested in [9] (for L_1 distances and

[⋆] Supported by grants MTM2009-08747 of the Spanish Ministry of Science and Innovation, SGR-2009-1122 of the Government of Catalonia, and INFRA-2010-262608 of the European Union.

binary variables) and [1] (for L_1, L_2 and L_∞ distances and with or without binary variables), as an alternative to the well known cell suppression problem (CSP) [2,13]. For a particular set of real instances, the quality of CTA solutions was observed to be higher than that obtained with CSP [5]. A recent discussion on these tabular data protection techniques can be found in the survey [4].

CTA was included in 2008 within a solution scheme for the protection of European structural business statistics. The resulting package [6] was developed within an Eurostat funded framework in collaboration with Statistics Germany and Statistics Netherlands [11]. Initially this CTA package was only linked to the state-of-the-art commercial solver Xpress, following the Eurostat requirements. Later, it was also hooked to Cplex, one of the best commercial solvers. The initial design of the code (meant only for the Xpress solver) forced to replicate and particularize according to the solver most of the model definition, solution, and solution retrieval interface routines. Although these routines are similar for most solvers, this procedure is cumbersome and error-prone. It also makes the maintenance of the CTA package difficult. This is specially relevant when alternative, efficient enough, open-source MILP solvers are available, and want to be tested by National Statistical Institutes (NSIs).

This work presents a new multisolver CTA package. The package is based on an object oriented design and it uses the Open Solver Interface (OSI) abstract base class, to be commented below, which allows easy integration with new available solvers. OSI is part of the COIN-OR project [14], sponsored by IBM. The multisolver CTA package is currently linked to five solvers: two commercial ones (Cplex and Xpress), and three open-source ones (Glpk, Cbc, and Symphony). Package Glpk is part of the GNU project of the Free Software Foundation. Packages Cbc and Symphony are also part of the COIN-OR project. The package is tested using a set of public and confidential real-world instances, which are classified into two groups of small and medium-sized problems. The computational results show that for small instances all solvers are consistently efficient, whereas for medium-sized instances there is a wider range of situations.

It is worth to note that, due to its object oriented design and the use of the OSI, this CTA package can be "easily" adapted for a multisolver version of the "continuous CTA" variant (i.e., without binary variables, thus formulating a simpler linear programming (LP) problem), which can be solved much more efficiently, at the expense of providing a lower quality solution. This approach was originally introduced in [1], and it has been suggested as the basis for on-line protection in tabular data servers [8]. Updating the current multisolver CTA package for this simpler LP problem is one of the forthcoming tasks.

The paper is organized as follows. Section 2 reviews the CTA MILP formulations for problems with both positive and negative protection levels. Section 3 describes the design and main features of the new package. Finally, Section 4 reports the computational evaluation between the five solvers with both standard instances found in the literature and real-world ones.

2 The CTA Model

A CTA instance is defined by the following parameters: (i) a set of cells $a_i, i \in \mathcal{N} = \{1, \ldots, n\}$, that satisfy some linear relations $Aa = b$ (a being the vector of a_i's); (ii) a lower and upper bound for each cell $i \in \mathcal{N}$, respectively l_{a_i} and u_{a_i}, which are considered to be known by any attacker; (iii) a set $\mathcal{S} = \{i_1, i_2, \ldots, i_s\} \subseteq \mathcal{N}$ of indices of sensitive cells; (iv) and a lower and upper protection level for each sensitive cell $i \in \mathcal{S}$, 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 structure of the table is defined by equations $Aa = b$, which can model any kind of table. We are not imposing $a_i \in \mathbb{Z}$, then fractional cell values can be obtained. The model is thus valid for magnitude tables, not frequency ones. However, in practice, it was observed that perturbed cell values obtained are integer without imposing $a_i \in \mathbb{Z}$ [1].

The purpose of CTA is to find the closest safe values $x_i, i \in \mathcal{N}$, according to some distance L, that make the released table safe. This involves the solution of the following optimization problem:

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

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

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

$z \in \mathbb{R}^n$ being the vector of deviations. Defining $z = z^+ - z^-$, (2) can be written for the L_1 distance as

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

$w \in \mathbb{R}^n$ 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 $y \in \mathbb{R}^s$ being the vector of binary variables associated to protections senses. When $y_i = 1$ the constraints mean $upl_i \leq z_i^+ \leq u_{zi}$ 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, in general, a (difficult) MILP.

If the problem has negative protection levels (i.e., $lpl_i < 0$ or $upl_i < 0$ for at least one cell i), the optimization model (3) is no longer valid. Problems with negative protection levels can be useful for the sequential protection of correlated tables [3]. The following alternative model, introduced in [3], may be used for these cases:

$$\begin{array}{ll}
\min\limits_{z^+,z^-,y} & \sum\limits_{i=1}^{n} w_i(z_i^+ + z_i^-) \\
\text{subject to} & A(z^+ - z^-) = 0 \\
& l_z \le z^+ - z^- \le u_z \\
& z_i^+ - z_i^- \ge upl_i y_i + l_{z_i}(1 - y_i) \quad i \in \mathcal{S} \\
& z_i^+ - z_i^- \le -lpl_i(1 - y_i) + u_{z_i} y_i \quad i \in \mathcal{S} \\
& (z^+, z^-) \ge 0 \\
& y_i \in \{0, 1\} \quad i \in \mathcal{S}.
\end{array} \tag{4}$$

The main difference between (4) and (3) is that (z^+, z^-) are not related to upper and lower protection deviations in (4), but they are just auxiliary variables to model the L_1 distance. As a result, model (4) is valid for any kind of instance, with either positive or negative protection levels. However, as shown in [3], it is less efficient than model (3), and then, (3) is preferred for problems with only positive protection levels. The multisolver CTA package described in the next section implements both models (3) and (4).

3 The Multisolver CTA Package

There are three main options for implementing a CTA package linked to several optimization solvers:

1. Developing ad hoc code for each particular solver. This was the choice for the previous CTA package [6], which was only linked to Cplex and Xpress. This is the most versatile and efficient option, but also the most time consuming and difficult to maintain, since future extensions in the model should be replicated in the particular code for each solver.
2. Using a generic modeling language (such as AMPL [10]), which is already hooked to several solvers. This option allows quick development of models, but it has two main drawbacks: good modeling languages are commercial and proprietary software (in some cases, linked to only one particular solver); and they are interpreted languages, so no compiler is available to generate an efficient executable. This option is mainly appropriate for testing models and prototypes.
3. A third option between the two above is using a generic model interface. This was the choice. In particular, we considered the open-source Open Solver Interface (OSI), a C++ generic base class to interface several optimization solvers. OSI is developed within the COIN-OR project [14].

Due to the use of the OSI, the new CTA package is written in C++ in an object oriented design. The optimization model is formulated as a generic OSI model, which can then be solved using some of the solvers interfaced to OSI. The current implementation is linked to Cplex, Xpress (commercial ones), Glpk, Cbc and Symphony (open-source ones). We remark that, although OSI provides several interface routines to communicate with the solvers, it does not offer the same flexibility than the option 1 above. Therefore, in practice, the CTA package also implements for each solver ad hoc code for some features (such as, for instance, restarting the optimization procedure after some time limit has been reached). C++ virtual functions [15] were used (for each solver) for procedures such as running the solver, closing the solver, getting full information about the solution, and applying the tool to repair infeasible instances [7]. The use of virtual functions allows clean separation between solvers, and an easy integration of new ones. Although virtual functions mean an overhead to the execution time [15], they are used only once—when the package has to deal with the particular solver—during the run.

The package implements several options which can be controlled by the user. Describing all of them is out of the scope of this work. Most of them are common to the previous CTA package, whose full details can be found in [6]. The most important parameters are:

– Solver to be used.
– Whether to stop at the first feasible solution.
– Several optimization tolerances (e.g., feasibility and integrality tolerance).
– Time limit of the optimization procedure (the default value is a very large time limit, i.e., one day of CPU time).
– Model to be used, either (3) or (4) [3].
– Whether to apply the "repair infeasibility" tool [7].
– Whether to make additive non-additive tables [3].
– The (percentage) optimality gap, which is computed as $gap = \frac{obj-lb}{1+|obj|} \cdot 100$, where obj is the objective function and lb is the lower bound provided by the solver. The default value is a 5% optimality gap.

To avoid biased results, the default values for these parameters were used for all the solvers in the computational evaluation of next section (but for the time limit, which was reduced). Of course, tuning some parameters it could be possible to speed up the solution process, but the goal of this work was to make a fair comparison between the five solvers.

4 Computational Results

We have considered a set of 33 instances, divided into two groups of "small" and "medium-sized" instances. Table 1 shows the main dimensions of these instances: number of cells (column n), number of sensitive cells (s), number of tabular linear relations (m), number of variables and constraints of formulation (3) ("vars" and "cons"), and percentage of binary variables ("%bin"). Instances with a large

Table 1. Dimensions of the test instances: number of cells n, number of sensitive cells s, number of tabular constraints m, number of variables and constraints of the optimization problems "vars." and "cons.", percentage of binary variables "%bin"

instance	n	s	m	vars.	cons.	%bin
Small instances						
idescat1	126	30	45	282	165	11.90
idescat2	126	27	45	279	153	10.71
idescat3	126	15	45	267	105	5.95
idescat4	126	12	45	264	93	4.76
idescat5	126	7	45	259	73	2.78
idescat6	126	35	45	287	185	13.89
MM140_m03m04	87	5	35	179	55	2.87
MM140_m04m05	87	5	35	179	55	2.87
MM140_m05m06	87	5	35	179	55	2.87
MM140_m06m07	87	5	35	179	55	2.87
MM140_m07m08	87	5	35	179	55	2.87
MM140_m08m09	87	5	35	179	55	2.87
MM140_m09m10	87	5	35	179	55	2.87
MM140_m10m11	87	5	35	179	55	2.87
MM140_m11m12	87	5	35	179	55	2.87
osorio	10201	7	202	20409	230	0.03
table7	624	17	230	1265	298	1.36
table8	1271	3	72	2545	84	0.12
targus	162	13	63	337	115	4.01
Medium-sized instances						
australia_ABS	24420	918	274	49758	3946	1.88
cbs	11163	2467	244	24793	10112	11.05
dale	16514	4923	405	37951	20097	14.91
destatis	5940	621	1464	12501	3948	5.23
hier13	2020	112	3313	4152	3761	2.77
hier16	3564	224	5484	7352	6380	3.14
sbs2008_C	4212	1135	2580	9559	7120	13.47
sbs2008_E	1430	382	991	3242	2519	13.36
table1	1584	146	510	3314	1094	4.61
table3	4992	517	2464	10501	4532	5.18
table4	4992	517	2464	10501	4532	5.18
table5	4992	517	2464	10501	4532	5.18
table6	1584	146	510	3314	1094	4.61
toy3dsarah	2890	376	1649	6156	3153	6.51

Table 2. Results with commercial solvers: objective function “obj”, optimality gap, primal optimality gap “pgap”, CPU time, and number of unprotected cells in the solution “unprot”, for each solver

	Cplex					Xpress				
instance	obj	gap	pgap	time	unprot	obj	gap	pgap	time	unprot
Small instances										
idescat1	286	0.00	0.00	0.01	0	286	0.00	0.00	0.09	0
idescat2	246	0.00	0.00	0.00	0	246	0.00	0.00	0.06	0
idescat3	150	0.00	0.00	0.01	0	150	0.00	0.00	0.07	0
idescat4	184	0.00	0.00	0.01	0	184	0.00	0.00	0.08	0
idescat5	104	0.00	0.00	0.01	0	104	0.00	0.00	0.08	0
idescat6	366	0.00	0.00	0.00	0	366	0.00	0.00	0.09	0
MM140_m03m04	12.0769	2.03	1.38	0.00	0	12.0769	2.03	1.38	0.07	0
MM140_m04m05	60.7774	4.79	3.93	0.01	0	58.3474	0.00	0.00	0.05	0
MM140_m05m06	17.5466	3.63	1.45	0.01	1	17.3426	2.56	0.36	0.06	0
MM140_m06m07	31.9702	0.03	-0.07	0.00	1	32.6932	2.07	2.07	0.07	0
MM140_m07m08	22.6275	5.04	0.22	0.00	0	22.6275	0.34	0.22	0.06	0
MM140_m08m09	46.737	4.73	4.18	0.01	0	46.737	4.18	4.18	0.09	0
MM140_m09m10	26.6791	2.82	-0.44	0.00	2	26.8003	0.00	0.00	0.08	0
MM140_m10m11	36.8517	0.86	0.54	0.00	0	36.6483	0.00	0.00	0.08	0
MM140_m11m12	30.6695	3.03	0.00	0.01	0	30.6694	0.00	0.00	0.07	0
osorio	13	0.00	0.00	0.83	0	13	0.00	0.00	0.37	0
table7	9.97027e+09	0.00	0.00	0.05	10	9.97027e+09	2.58	0.00	0.06	10
table8	439	0.00	0.00	0.05	0	439	3.64	0.00	0.13	0
targus	1.07114e+06	0.57	0.19	0.02	0	1.11833e+06	4.41	4.41	0.05	0
Medium-sized instances										
australia_ABS	632	2.09	2.09	5.13	2	631	3.96	1.94	0.77	2
cbs	0	0.00	0.00	0.77	0	0	0.00	0.00	1.18	2
dale	39	0.00	0.00	32.32	25	19	0.00	-100.00	2.27	37
destatis	2.35352e+08	3.67	2.22	81.42	0	2.36564e+08	2.72	2.72	88.95	0
hier13	4.34835e+08	4.93	4.68	954.11	0	4.36127e+08	4.96	4.96	497.76	0
hier16	5.1264e+08	59.04	40.52	10001.30	0	5.70171e+08	66.02	46.52	10000.00	0
sbs2008_C	313876	1.99	1.99	107.06	1	—	—	—	—	—
sbs2008_E	107669	2.93	2.93	3.32	0	—	—	—	—	—
table1	2.87934e+13	3.67	0.91	164.47	26	2.93396e+13	2.76	2.76	2.14	26
table3	1.20927e+12	21.31	-7.16	10000.30	42	1.36333e+12	4.95	4.95	621.52	61
table4	1.02322e+10	21.16	-10.44	10000.30	49	1.18853e+10	4.92	4.92	665.38	73
table5	9.8721e+06	12.81	-7.91	10000.90	42	1.11682e+07	4.61	4.61	708.83	48
table6	2.83577e+10	4.69	4.69	0.61	15	2.80496e+10	3.77	3.65	0.77	35
toy3dsarah	1.4784e+15	100.00	100.00	0.04	0	—	—	—	—	—

Table 3. Results with open-source solvers: objective function "obj", optimality gap, primal optimality gap "pgap", CPU time, and number of unprotected cells in the solution "unprot", for each solver

	Glpk					Cbc				Symphony			
instance	obj	gap	pgap	time	unprot	obj	pgap	time	unprot	obj	pgap	time	unprot
Small instances													
idescat1	286	0.00	0.00	0.00	0	286	0.00	0.01	0	286	0.00	0.00	0
idescat2	246	0.00	0.00	0.01	0	246	0.00	0.01	0	246	0.00	0.00	0
idescat3	150	0.00	0.00	0.00	0	150	0.00	0.01	0	150	0.00	0.01	0
idescat4	184	0.00	0.00	0.00	0	184	0.00	0.01	0	184	0.00	0.01	0
idescat5	104	0.00	0.00	0.00	0	104	0.00	0.01	0	104	0.00	0.01	0
idescat6	366	0.00	0.00	0.01	0	366	0.00	0.01	0	366	0.00	0.01	0
MM140_m03m04	11.908	0.09	0.09	0.01	0	11.908	0.09	0.00	0	11.908	0.09	0.00	0
MM140_m04m05	58.3474	72.25	0.00	0.01	0	58.3474	0.00	0.00	0	58.3474	0.00	0.00	0
MM140_m05m06	17.3426	0.36	0.36	0.00	0	17.3426	0.36	0.01	0	17.3426	0.36	0.00	0
MM140_m06m07	32.0631	0.21	0.21	0.01	0	32.0631	0.21	0.01	0	31.9601	-0.10	0.00	1
MM140_m07m08	22.6275	0.22	0.22	0.00	0	22.6275	0.22	0.00	0	22.6275	0.22	0.01	0
MM140_m08m09	44.8024	44.09	0.13	0.01	0	44.8024	0.13	0.00	0	44.8024	0.13	0.01	0
MM140_m09m10	26.8003	0.21	0.00	0.00	0	26.8003	0.00	0.01	0	26.716	-0.30	0.00	4
MM140_m10m11	36.8517	0.68	0.54	0.01	0	36.6483	0.00	0.01	0	36.5722	-0.20	0.00	4
MM140_m11m12	30.6694	0.16	0.00	0.00	0	30.6694	0.00	0.01	0	30.6694	0.00	0.01	0
osorio	13	7.14	0.00	2.77	0	13	0.00	4.83	0	13	0.00	2.19	0
table7	1.00451e+10	0.74	0.74	0.05	2	1.00118e+10	0.41	0.24	0	9.97027e+09	0.00	0.03	10
table8	439	52.73	0.00	0.06	0	439	0.00	0.06	0	439	0.00	0.04	0
targus	1.08855e+06	2.06	1.79	0.02	0	1.07114e+06	0.19	0.16	0	1.07114e+06	0.19	0.03	0
Medium-sized instances													
australia_ABS	1652	83.06	62.51	9968.72	10	1063	41.75	11301.70	0	73342	99.15	9988.38	0
cbs	3.88907e+06	100.00	100.00	9973.26	1	0	0.00	1063.11	0	1.88788e+11	100.00	9960.58	0
dale	49828	100.00	99.92	9960.96	27	256	84.44	10423.50	0	0	-3900.00	393.86	36
destatis	2.34543e+08	22.79	1.88	9997.92	0	2.63593e+08	12.69	10039.50	0	3.37156e+08	31.74	99.06	5
hier13	4.36127e+08	4.96	4.96	113.16	0	4.34835e+08	4.68	5293.69	0	4.60856e+08	10.06	33.95	0
hier16	5.31621e+08	42.64	42.64	9985.09	0	5.57426e+08	45.29	10028.20	0	6.50722e+08	53.14	9961.52	0
sbs2008_C	314745	7.85	2.26	9965.97	0	8.48981e+22	100.00	10032.60	26	321191	4.22	9968.03	1
sbs2008_E	—	—	—	—	—	107999	3.22	10141.40	0	111740	6.46	6.99	0
table1	2.89023e+13	4.97	1.28	9.95	1	2.90983e+13	1.95	10150.80	0	3.04182e+13	6.20	6.80	13
table3	1.23277e+12	20.43	-5.11	9994.11	0	1.73912e+12	25.49	10020.70	0	1.67018e+12	22.41	157.16	17
table4	1.02468e+10	20.78	-10.29	9961.10	0	1.33087e+10	15.09	10025.10	0	1.12156e+10	-0.76	9960.99	13
table5	1.03974e+07	19.23	-2.46	9992.39	87	6.96616e+07	84.71	10045.00	14	—	—	—	—
table6	2.82351e+10	4.99	4.28	40.97	3	2.82492e+10	4.33	10126.50	4	2.81623e+10	4.03	3805.12	12
toy3dsarah	—	—	—	—	—	—	—	—	—	—	—	—	—

number of cells, but only a few of them sensitive, such as “osorio”, “table7” and “table8” are classified as small. Instances “australia_ABS” (provided by the Australian Bureau of Statistics), “idescat*” (provided by the Statistical Institute of Catalonia), “MM140_m*m*” (provided by Eurostat), “destatis” (provided by the German Federal Statistical Office), and “sbs2008_*” (provided by Eurostat) are confidential real-world ones; the remaining ones are public and have been previously used in the literature [1,3,4].

These 33 instances were run with the five solvers in the CTA package, the commercial Cplex and Xpress, and the open-source Glpk, Cbc and Symphony. A time limit of 10000 seconds was set for all the executions. The default values were used for the remaining parameters, the most relevant being the optimality gap (5%) and the solution of model of formulation (3) (instead of model (4)). All runs were carried out on a Fujitsu Primergy RX300 server with 3.33GHz Intel Xeon X5680 CPUs and 144 GB of RAM, under a GNU/Linux operating system (Suse 11.4), without exploitation of parallelism capabilities. Because of their relatively small dimensions, these instances can also be solved in a much smaller laptop or desktop PC.

Tables 2 and 3 show the results obtained with, respectively, the commercial and open-source solvers, again differentiating between small and medium-sized instances. For the commercial solvers Cplex and Xpress, and the open-source solver Glpk, the following information is provided: objective value of the solution found (column “obj”), percentage optimality gap of the solution found (“gap”), the percentage “primal gap” (“pgap”, defined below), the CPU time needed for the solution (“time”), and the number of unprotected cells (“unprot”). For the open-source solvers Cbc and Symphony the same information is provided excluding the “gap” column (since the information about the lower bound of the solution can not be currently obtained from neither Cbc nor Symphony). Executions that reported no feasible solution are marked with “—”. The “primal gap” is computed by comparing the objective function obj of each solver with the best lower bound found by the five solvers lb_{best}, i.e., $pgap = \frac{obj - lb_{best}}{1+|obj|} \cdot 100$, providing a global measure of the quality of the solution found (instead of the local measure given by “gap”). Unprotected cells are usually obtained when large or big-M values appear in the constraints involving binary variables of model (3). For instance, if u_{z_i} is very large (e.g., 10^9) and $y_i = 10^{-8} \approx 0$ then the constraint $z_i^+ \leq u_{z_i} y_i$ imposes $z_i^+ \leq 10$ (instead of the right constraint $z_i^+ \leq 0$), likely resulting in an unprotected cell. Big values (e.g., big cell bounds) should be avoided by users of CTA. Alternative approaches as logical “indicator constraints” (available in Cplex and Xpress) partially fix this issue, but significantly increase the solution time.

The following comments can be drawn from tables 2 and 3:

– As mentioned above, some executions provide a number of unprotected cells. This more often happens for the medium-sized than for the small instances. The number of executions with unprotected cells for each solver was: Cplex 12; Xpress 9; Glpk 7; Cbc 3; and Symphony 11. Surprisingly, this number of underprotected cells is larger for the two commercial solvers than for the

open-source ones. Though there is not a clear explanation, it might be due to the more aggressive heuristics used by commercial solvers.
- The "pgap" of some executions is negative, i.e., the objective function found by the solver is less than the best (i.e., highest) lower bound by the five solvers. Obviously, this should never happen in theory. However, we observe that this occurs in executions with unprotected cells, i.e, by violating the protection requirement of some sensitive cells the solver may provide objective functions below the lower bound obtained by another solver. Negative "pgaps" are mainly obtained for the medium-sized instances. The number of executions with negative "pgaps" for each solver was: Cplex 6; Xpress 2; Glpk 3; Cbc 0; and Symphony 6. The worst cases were executions with Xpress and Symphony for instance dale, which provided pgaps of -100% and -3900%, respectively.
- Some executions present strangely large gaps at optimal solutions, (computed without exhausting the time limit). One is instance toy3dsarah with Cplex (gap of 100% in the optimal solution found). In this case no lower bound was computed, since this problem was heuristically solved by Cplex. However, the other four solvers reported this instance as infeasible with the default infeasibility tolerances. Large gaps are also provided by Glpk at the optimal solutions of instances table8, MM140_m04m05, MM140_m08m09: the cause seems to be that the solver did not internally update the lower bound when the branch-and-cut tree was emptied.
- About the efficiency, Cplex exhausted the time limit in 4 executions (and it provided a solution for all the instances, though some unprotected, as discussed above); Xpress reached the time limit in 1 run (but it did not solve 3 instances); Glpk in 9 (and it did not solve 2 instances); Cbc in 11 (with 1 unsolved instance); and Symphony in 5 (with 2 unsolved instances). All the executions that reached the time limit were for medium-sized instances.

5 Conclusions

From the computational results of this work we can conclude that for small instances, both commercial and open-source solvers behave similarly. On the other hand, there is a slight advantage for commercial solvers in larger instances, which increases with the size of the problem. However, models (3) and (4) are difficult MILP problems. If the values of the binary variables y are a priori fixed, formulations (3) and (4) become continuous optimization problems, which can be solved much more efficiently, at the expense of providing a lower quality solution. This approach was introduced in [1], and it has been suggested as the basis for on-line protection in tabular data servers [8]. Updating the multisolver CTA package of this work for this simpler problem is part of the future tasks to be done. The gap between the commercial and open-source solvers is expected to be reduced for the continuous CTA problem.

Acknowledgments. We thank Mr. Xavier Jiménez-Marimon for his help with the integration of the CTA package with the OSI. We also thank Idescat for providing and giving permission to use the "idescat*" instances in this work.

References

1. Castro, J.: Minimum-distance controlled perturbation methods for large-scale tabular data protection. European Journal of Operational Research 171, 39–52 (2006)
2. Castro, J.: A shortest paths heuristic for statistical disclosure control in positive tables. INFORMS Journal on Computing 19, 520–533 (2007)
3. Castro, J.: Extending controlled tabular adjustment for non-additive tabular data with negative protection levels. Statistics and Operations Research Transactions–SORT 35, 3–20 (2011)
4. Castro, J.: Recent advances in optimization techniques for statistical tabular data protection. European Journal of Operational Research 216, 257–269 (2012)
5. Castro, J., Giessing, S.: Testing variants of minimum distance controlled tabular adjustment. In: Monographs of Official Statistics, pp. 333–343. Eurostat-Office for Official Publications of the European Communities, Luxembourg (2006)
6. Castro, J., González, J.A., Baena, D.: User's and programmer's manual of the RCTA package. Technical Report DR 2009-01, Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya (2009)
7. Castro, J., González, J.A.: A Tool for Analyzing and Fixing Infeasible RCTA Instances. In: Domingo-Ferrer, J., Magkos, E. (eds.) PSD 2010. LNCS, vol. 6344, pp. 17–28. Springer, Heidelberg (2010)
8. Castro, J., González, J.A.: Present and future research on controlled tabular adjustment. In: Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality (2011), `http://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/2011/48_Castro-Gonzalez.pdf`
9. Dandekar, R.A., Cox, L.H.: Synthetic tabular Data: an alternative to complementary cell suppression, manuscript, Energy Information Administration, U.S. (2002)
10. Fourer, R., Gay, D.M., Kernighan, D.W.: AMPL: A Modeling Language for Mathematical Programming. Duxbury Press (2002)
11. Giessing, S., Hundepool, A., Castro, J.: Rounding methods for protecting EU-aggregates. In: Eurostat Methodologies and Working Papers. Worksession on Statistical Data Confidentiality, pp. 255–264. Eurostat-Office for Official Publications of the European Communities, Luxembourg (2009)
12. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Lenz, R., Naylor, J., Schulte-Nordholt, E., Seri, G., de Wolf, P.P.: Handbook on Statistical Disclosure Control (v. 1.2), Network of Excellence in the European Statistical System in the field of Statistical Disclosure Control (2010), `http://neon.vb.cbs.nl/casc/SDC_Handbook.pdf`
13. Kelly, J.P., Golden, B.L., Assad, A.A.: Cell suppression: disclosure protection for sensitive tabular data. Networks 22, 28–55 (1992)
14. Lougee-Heimer, R.: The Common Optimization INterface for Operations Research. IBM Journal of Research and Development 47, 57–66 (2003)
15. Stroustrup, B.: The C++ Programming Language. Addison-Wesley (1997)