Using BCD-CTA for difficult tables: a practical experiment with a real Eurostat table [1]

Jordi Castro*, José A. González*, Anco Hundepool **

* Department of Statistics and Operations Research, Universitat Politècnica de Catalunya Jordi Girona 1–3, 08034 Barcelona, Catalonia. (jordi.castro@upc.edu, jose.a.gonzalez@upc.edu)

** Wulplaan 17, 2261 DG Leidschendam, The Netherlands. (hundepool@ziggo.nl)

Abstract. CTA is a post-tabular perturbative approach for statistical disclosure control. Its purpose is to compute the closest safe table to the original data, using some distance. Sensitive cells are adjusted either upwards or downwards (binary decision), and the resulting cells have to be accordingly (and minimally) modified to preserve marginals. For real and large tables, CTA may result in a difficult mixed integer linear problem for some weights in the objective function. In those situations the Block Coordinate Descent (BCD) heuristic for CTA—which is included in the Tau-Argus CTA distribution—may be used to quickly obtain a feasible, hopefully close to optimality, solution. We present a practical experiment using a large and difficult real-world table from Eurostat. We will show that, for unitary weights, while the standard CTA can not obtain a solution in about half an hour, the BCD-CTA approach provides a solution in few seconds.

1 Introduction and motivation

The protection of confidentiality of tables published at the EU-level is a joint concern for the National Statistical Institutes (NSIs) and Eurostat. The normal practice is that the NSIs collect and process the information of each member state. This includes also the confidentiality protection, when publishing the information. For tabular data the primary unsafe cells are based on the p%-rule or the (n, k)-rule (Hundepool et al., 2012). There is a growing number of member states that is using the p%-rule. The second step, finding the necessary secondary cells to fully protect the table, is a computational complex task Castro (2011) which is often done with the software τ-Argus (Wolf et al., 2014). This is fine at the national level. After this process the NSIs send their tables to Eurostat.

[1] This work has been supported by grant MTM2012-31440 of the Spanish research program, and partially by project DwB INFRA-2010-262608 of the EU FP7.

It is the task of Eurostat to compile the complete European table, including the EU-aggregates. In order to allow Eurostat to compute these EU-aggregates, each NSI send the complete table with flags for the primary and secondary suppressed cells. Eurostat has the obligation to guarantee the confidentiality of the national data. It is obvious that if one country did suppress a certain cell, Eurostat cannot publish the EU-aggregate. Even if more countries suppress a cell and there is one dominating confidential country the EU-aggregate cannot be published as well. This is a very unsatisfying situation. Alternatively waiting for all 27 countries to complete their table and protecting the big European table in one big run will not work either, because each Member State has to wait for the last one before they can publish their own table.

The solution suggested by Giessing et al. (2009) was to compute rounded figures for the EU-cells at risk. The rounding base for each cell should be large enough to safeguard the national confidential cells. It is often much better to have a rounded EU-aggregate that no figure at all. It is to be expected that more cells have to be protected in the smaller Member States, as they often have only one or a few enterprises in a certain cell. While protecting the smaller national figures, the rounded EU-aggregate can still be a reasonable informative figure.

In the process of computing the necessary uncertainty to be added to the EU-aggregates we use the controlled tabular adjustment (CTA) procedure, as described in Giessing et al. (2009). In this work we applied this procedure to a real-life dataset for the EU structural business statistics (SBS). It is a three-dimensional table. The first dimension is the EU-member state (27 plus the EU-total) without a hierarchy; the second dimension is a NACE classification with a hierarchy and in total 120 codes; the third dimension is a size-class (5 codes plus a total). This amounts to a total of $28 \cdot 120 \cdot 6 = 20160$ cells. For a plain three-dimensional tables the number of constraints would be $(28 + 6) \cdot 120 + 28 \cdot 6 = 4248$. However, due to the hierarchy of the NACE variables, which implies extra linear cells relations, the total number of constraints is 8280.

When running the tests it turned out that CTA took an unacceptable long running time for certain problems. This had led to the use of the Block Coordinate Descent (BCD) heuristic for the CTA procedure (González and Castro, 2011), which is included in the most recent τ-Argus distribution. The standard MILP-CTA and BCD-CTA will be outlined in Sections 2 and 3 of this paper, respectively. However also the type of cost function gave a remarkable difference in running time. These computational experiments will be reported in Section 4 of the paper.

2 Outline of minimum distance MILP-CTA

CTA (Dandekar and Cox, 2002; Castro, 2006) is a post-tabular approach which looks for the closest safe table to the original unsafe table. CTA achieves disclosure

limitation by either increasing or decreasing by at least a certain amount (*protection level*) the cell values of a subset of sensitive cells, and then adjusting the rest of cells to preserve some desired constraints. CTA is formulated as a mixed integer linear programming (MILP), whose parameters are:

- 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), and a vector $w \in \mathbb{R}^n$ of positive weights for the deviations of cell values.
- A lower and upper bound for each cell $i = 1, \ldots, n$, respectively l_{x_i} and u_{x_i}, which are considered to be known by any attacker. If no previous knowledge is assumed for cell i $l_{x_i} = 0$ ($l_{x_i} = -\infty$ if $a \geq 0$ is not required) and $u_{x_i} = +\infty$ can be used.
- A set $\mathcal{S} = \{i_1, i_2, \ldots, i_s\} \subseteq \{1, \ldots, n\}$ of indices of s confidential cells.
- A lower and upper protection level for each confidential cell $i \in \mathcal{S}$, respectively lpl_i and upl_i, such that the released values $x_i, i = 1, \ldots, n$, satisfy either $x_i \geq a_i + upl_i$ or $x_i \leq a_i - lpl_i$.

CTA attempts to find the closest 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:

$$\begin{aligned} \min_{x} \quad & ||x - a||_\ell \\ \text{subject to} \quad & Ax = b \\ & l_x \leq x \leq u_x \\ & x_i \leq a_i - lpl_i \text{ or } x_i \geq a_i + upl_i \quad i \in \mathcal{S}. \end{aligned} \tag{1}$$

Problem (1) can also be formulated in terms of deviations from the current cell values. Defining $z = x - a$, $l_z = l_x - a$, $u_z = u_x - a$, using the ℓ_1 distance weighted by w, and introducing variables $z^+, z^- \in \mathbb{R}^n$ so that $z = z^+ - z^-$ and $|z| = z^+ + z^-$, the final MILP model for CTA is:

$$\min_{z^+, z^-, y} \quad \sum_{i=1}^{n} w_i (z_i^+ + z_i^-) \tag{2a}$$

$$\text{subject to} \quad A(z^+ - z^-) = 0 \tag{2b}$$

$$0 \leq z^+ \leq u_z, \quad 0 \leq z^- \leq -l_z \tag{2c}$$

$$y \in \{0, 1\}^s \tag{2d}$$

$$\left.\begin{array}{ll} upl_i\, y_i & \leq z_i^+ \leq u_{z_i} y_i \\ lpl_i (1 - y_i) & \leq z_i^- \leq -lz_i (1 - y_i) \end{array}\right\} i \in \mathcal{S} \tag{2e}$$

Constraints (2b) impose feasibility of the published perturbed table. Constraints (2c) guarantee perturbations are within allowed bounds. Constraints (2d)–(2e) force the new table is safe. When $y_i = 1$ the constraints mean $upl_i \leq z_i^+ \leq u_{z_i}$ and $z_i^- = 0$, thus the protection sense is "upper"; when $y_i = 0$ we get $z_i^+ = 0$ and $lpl_i \leq z_i^- \leq -l_{z_i}$, thus the protection sense is "lower".

3 Outline of minimum distance BCD-CTA

Coordinate descent is a family of optimization algorithms that successively optimize along coordinate directions. They were popular in the 1980s and 1990s, but, due to its simplicity and low computational cost, they recently gained reputation for approximate solutions in big-data problems (see Wright (2015) for a recent survey).

When instead of optimizing over a coordinate, or single variable, they optimize on a block of variables, they are named block coordinate descent (BCD). Therefore, BCD solves a sequence of subproblems, each of them optimizing the objective function over a subset of variables while the remaining variables are kept fixed. This is iteratively repeated until no improvement in the objective function is achieved, or some other end criteria is met (like a time limit). Convergence of this algorithm is only guaranteed for convex problems where each optimization subproblem has a unique optimizer (Bersekas, 1999, Prop. 2.7.1). Although MILP problems are not convex, and thus they do not guarantee convergence, BCD usually behaves well in practical complex applications, and it can be used as a heuristic approach.

BCD was used in González and Castro (2011) to efficiently obtain approximate solutions to CTA problems. BCD CTA may provide good approximate solutions by optimizing at each iteration the protection direction (either "downward" or "upward") of a subset of sensitive cells, and the deviations for all the cells. The protection directions of the remaining sensitive cells are kept constant at the optimal values of previous iterations. Note that continuous variables of the problem (the deviations for all the cells) are never fixed; unlike the standard BCD approach, blocking and fixing is only performed for the binary variables. Partitioning the binary variables y of (2a)–(2e) into k blocks, and denoting $y^{j,i}$ as the fixed values of block j at inner iteration i, the algorithm is roughly as follows:

Step 0 Initialization. Set outer iteration counter: $t = 0$. Set initial values, hopefully feasible, to y.

Step 1 $t = t + 1$. Set inner iteration counter $i = 0$.
Divide y into k blocks: $y = \{y^{1,i}, \ldots, y^{k,i}\}$, not necessarily of the same size.

Step 1.1 $i := i + 1$. Solve (2a)–(2e) with respect to block $y^{i,i}$, taking into account that $y^{j,i}$ is fixed for $j \neq i$.
Let $y^{i,i+1} = (y^{i,i})^*$ (the point at the optimum). Let $y^{j,i+1} = y^{j,i}$ for $j \neq i$.

Step 1.2 If $i < k$ go to Step 1.1.

Step 2 Check for end conditions: if apply, stop, and return the current best solution. Otherwise, go to Step 1

The above algorithm has been recently implemented and added to τ-Argus in the scope of the Data without Boundaries INFRA-2010-262608 EU project. Note that the original problem (2a)–(2e) is solved if only one block of variables is considered. Therefore, although BCD-CTA is a heuristic, it is easily switched to an optimal approach for CTA by setting $k = 1$ at Step 1 for some advanced t. The subproblems of Step 1.1 may be solved by any MILP method. The τ-Argus implementation of BCD-CTA allows the solution of these subproblems by several free and commercial solvers.

One of the drawbacks of BCD-CTA is that it may not obtain a feasible solution unless the initial values of y (protection directions) are feasible. Several strategies are available in the τ-Argus implementation of BCD-CTA to compute such an initial feasible point (see González and Castro (2011)) for details). To overcome this drawback, BCD has been recently combined with another heuristic named fix-and-relax (FR) for CTA: FR computes a good initial feasible point, BCD improves on it (Baena et.al, 2015).

4 Computational results

For the computational experiments, we solved the three-dimensional Eurostat SBS table with both MILP-CTA and BCD-CTA, considering three different weights: $w_i = 1$, $w_i = 1/a_i$ and $w_i = 1/\sqrt{a_i}$. The runs were carried out on a PC with an I7 CPU at 3.40 GHz, using τ-Argus under Windows 8. Of the five available solvers in the CTA τ-Argus distribution (two commercial, three free), we used CPLEX. The results are summarized in Table 1; the meaning of its columns is provided below.

Initially, this instance was solved with $w_i = 1$ and MILP-CTA. As shown in Table 1, it took 7692 seconds (for a suboptimal solution, the procedure was stopped after 2 hours of CPU). This indeed motivated using BCD-CTA in this difficult case: BCD-CTA provided a "decent" suboptimal solution in only 19 seconds. This same instance with the two other weights resulted to be (unexpectedly) much easier: it took 85 and 179 seconds with MILP-CTA, and 16 seconds for the two weights with BCD-CTA.

In order to have an idea of the results of the CTA run, we computed the average deviation of the cells that have been modified; this information is reported in columns "$\bar{x}$ CTA" of Table 1. It is worth remarking that, in a strict sense, this measure can only be used to compare the results between MILP-CTA and BCD-CTA with $w_i = 1/a_i$, since the objective function minimized with those weights coincides with the average cell deviation. However we chose that measure for its simplicity. When

Table 1: Results for MILP-CTA and BCD-CTA with three different weights

	MILP-CTA			BCD-CTA		
w_i	CPU	$\bar{x}$ CTA	$\bar{x}$ published	CPU	$\bar{x}$ CTA	$\bar{x}$ published
1	7692	38933	64490	19	46443	91977
$1/a_i$	85	46904	89451	16	45900	93617
$1/\sqrt{a_i}$	179	40717	65297	16	42785	85098

$w_i = 1/a_i$, BCD-CTA did a good job: it provided a solution with less average deviations in less time. For the other two weights, the solutions of MILP-CTA provided lower values for this measure (however, we have to remind this may be just by chance, since with $w_i = 1$ and $w_i = 1/\sqrt{a_i}$ we are not minimizing the average deviations). To have a clearer picture on the size of the deviations, Figure 1 plots a graphical representation of the absolute deviations after one minute of BCD-CTA followed by two hours of MILP-CTA with weights $w_i = 1$. The plot shows a 28 rows times 720 columns matrix. Each entry of this matrix is associated to a cell table: the 28 rows are associated to the categories of the "member state" variable of the three-dimensional table, while the 720 columns are associated to the Cartesian product of the 120 NACE categories by the 6 categories of the size-class variable. Sizes of deviations are represented by colors, as reported in the legend. Member states are sorted by absolute deviations (the higher in the plot, the larger the deviations). We clearly see that there is one "member state" category with significantly larger absolute deviations than the others: this category is the total for the 27 member states. If the plot represented relative deviations, this category would not likely be the first one in this ranking.

Of course in this example we are not really interested in the results of the CTA solution but the published rounded EU-table. The average deviation of the published cells are given by columns "$\bar{x}$ published" of Table 1. As it is shown $w_i = 1/\sqrt{a_i}$ performed better than the other weights in this instance. And the best combination was $w_i = 1/\sqrt{a_i}$ with MILP-CTA.

5 Conclusions and future work

There is of course a lot to be said on the choice of the objective function. A certain deviation for a large cell high in the hierarchy can be less harmful than the same deviation for a small cell down in the hierarchy, but on the other hand we have similar EU-tables where the cells in the lower hierarchy are considered very important. Therefore, a weight denoting the level in the hierarchy could be a valuable option; indeed this was also the conclusion reached in Castro and Giessing (2006), where weights $w_i = 1/a_i^{1/\gamma_i}$ (γ_i depending on the cell hierarchy) provided the best results.

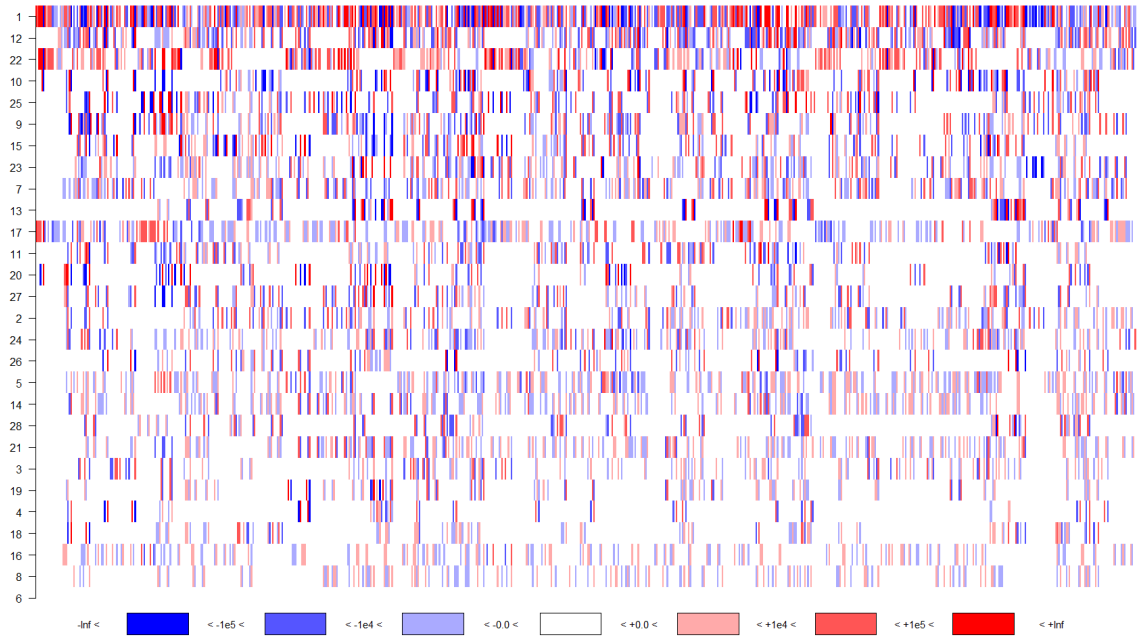

Figure 1: Graphical representation of the absolute cell deviations after one minute of BCD-CTA followed by two hours of MILP-CTA with weights $w_i = 1$. Each entry of this matrix is associated to a cell table: the 28 rows are associated to the categories of the "member state" variable of the three-dimensional table, while the 720 columns are associated to the Cartesian product of the 120 NACE categories by the 6 categories of the size-class variable. Sizes of deviations are represented by colors according to the legend.

The conclusion is that the alternatives for the weight function have a large implication. A careful analysis is part of the future tasks to be done. BCD-CTA has an enormous gain in computing efficiency, but it has a price. Nevertheless these options should be included in the software for Eurostat, since it may allow the solution of very large and intractable tables by other approaches.

From an optimization point of view, among the future task we find to understand why the behaviour of the MILP solver changes so drastically with different objective functions.

References

Baena, D., Castro, J., González, J.A. (2015). Fix-and-relax approaches for controlled tabular adjustment, *Computers & Operations Research*, 58, 41–52.

Bertsekas, D.P. (1999), *Nonlinear Programming, 2nd ed.*, Athena Scientific, Belmont, USA.

Castro, J. (2006). Minimum-distance controlled perturbation methods for large-scale tabular data protection, *European Journal of Operational Research*, 171, 39–52.

Castro, J. (2011). Recent advances in optimization techniques for statistical tabular data protection, *European Journal of Operational Research*, 216, 257–269.

Castro, J., Giessing, S. (2006). Testing variants of minimum distance controlled tabular adjustment, in *Monographs of Official Statistics. Work session on Statistical Data Confidentiality*, Eurostat-Office for Official Publications of the European Communities, Luxembourg, 333–343.

Dandekar, R.A., and Cox, L.H. (2002). Synthetic tabular data: an alternative to complementary cell suppression, manuscript, Energy Information Administration, U.S. Department of Energy.

Giessing, S., Hundepool, A., and Castro, J. (2009). Rounding methods for protecting EU-aggregates, in *Worksession on statistical data confidentiality. Eurostat methodologies and working papers*, Eurostat-Office for Official Publications of the European Communities, Luxembourg, 255–264.

González, J.A., and Castro, J. (2011) A heuristic block coordinate descent approach for controlled tabular adjustment, *Computers & Operations Research* 38, 1826–1835.

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Schulte Nordholt, E., Spicer, K., and Wolf, P.P. de (2012), *Statistical Disclosure Control*, Wiley, Chichester, United Kingdom.

Wolf. P.P. de, Hundepool, A.J., Giessing, S., Salazar, J.J., and Castro, J. (2014), *τ-ARGUS User's manual*, Statistics Netherlands, The Hague.

Wright, S.J. (2015). Coordinate descent algorithms, *Mathematical Programming*, 151, 3–34.