Working Paper
ENGLISH ONLY

UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE (UNECE)

CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION

STATISTICAL OFFICE OF THE EUROPEAN UNION (EUROSTAT)

Joint UNECE/Eurostat work session on statistical data confidentiality
(Ottawa, Canada, 28-30 October 2013)

Topic (iv): The trade-off between quality, utility and privacy

Secondary Cell Suppression in Tabular Data: A Comparison of Methods Implemented in τ-Argus

Prepared by Maxime Bergeat, French National Institute for Statistics and Economic Studies (Insee), France (maxime.bergeat@insee.fr)

Secondary cell suppression in tabular data: a comparison of the methods implemented in τ-Argus

Maxime Bergeat*

* French National Institute for Statistics and Economic Studies (Insee), Paris
maxime.bergeat@insee.fr

Abstract. They are three principal ways to deal with a secondary cell suppression problem in τ-Argus. The Hypercube algorithm gives a fast solution that may cause overprotection. The Optimal method solves a problem of Mixed Integer Linear Programming and the Modular approach enables to find more quickly a near-optimal solution for hierarchical tables. This paper discusses the advantages and disadvantages of these three methods in terms of quality and usability. It carefully examines the time needed by algorithms to converge against the loss of information contained in suppressed cells. Typical cases are tested on real data sets: one single explanatory variable, two hierarchical variables, sets of linked tables... Some guidelines are then proposed to find the best trade-off between quality of released data and time consumption. This paper concludes with the expected evolution of SDC software and what it will involve about the choice of a secondary cell suppression method.

1 Introduction

Lots of tables are disseminated by national statistics institutes (NSI). In order to keep respondents confident, NSI must take care of applying confidentiality treatments to avoid disclosure risk. Several techniques may be applied: recodings, rounding procedures, cell suppression... The cell suppression problem consists first in an identification of sensitive cells (primary suppression). Generally margins are included in published data and other cells must be suppressed to protect against disclosure by differencing. This secondary suppression leads to a loss of information that can be important for very large or detailed tables. The best suppression pattern aims at minimizing the cost of suppressed cells. This cost is generally set to the cell values. The software τ-Argus enables to detect sensitive cells and offers four methods to choose secondary suppressed cells. A brief description of these methods is given in Section 2. Three methods are compared in the empirical study introduced in Section 3. Quantitative results of this study are presented in Section 4. As a

conclusion of this paper, a discussion is led in Section 5 and some guidelines are proposed for typical "real-life" instances.

2 Algorithms implemented in τ-Argus

Four algorithms are implemented in the τ-Argus software in order to solve the secondary cell suppression problem (CSP). More details about implementation and theoretical background of the methods can be found in Hundepool and al. (2011).

Fischetti and Salazar (2000) propose a methodology where they formulate the CSP as a global minimization problem: the sum of the cost of suppressed cells is minimized under constraints of table additivity and respect of protection intervals. A set of feasible solutions is computed and is successively reduced. The convergence of this cut-and-branch algorithm can be very long: the user has the possibility to stop the execution after a predefined time and accept the reached solution.

The Modular approach presented in De Wolf (2002) is a heuristic used for hierarchical tables. The initial table is subdivided into small non-hierarchical tables where the optimal algorithm is applied. The information loss is optimal in each sub-table but the global cost of suppressed cells is not necessarily minimum.

These two approaches solve Mixed Integer Linear Programming (MILP) problems and rely on the use of a commercial solver.

Castro (2002) describes an algorithm based on network flows. The τ-Argus implementation is only applicable for 2-dimensional tables with one hierarchical variable. The network method brings a fast solution to the CSP generally close to the optimal one. However the network flow can only be used on very specific cell suppression problems.

The Hypercube method synthesized in Giessing and Riepsilber (2002) is a heuristic that finds very quickly a solution even for very detailed and high-dimensional tables. After having subdivided the initial table into sub-tables without substructure, each primary unsafe cell is protected successively by finding the best suppression pattern in the form of an hypercube: a sensitive cell is sufficiently protected if it is contained in an hypercube where the corner points are suppressed cells. This method often leads to over-protection because the set of suppression patterns considered is restricted.

License cost τ-Argus is a publicly available freeware. To use the Optimal and Modular approaches τ-Argus calls a commercial solver. The Hypercube and Network methods are free.

Negative cell values Generally when dealing with economic variables data producers consider only positive variables. In some cases variables may be negative (and therefore the resulting cell totals) that can cause some problems in the formulation

of minimization problems. In the current version of τ-Argus, only Hypercube and Modular techniques address this issue.

Singletons Protection of cells with a single respondent needs extra care. If there are in a same row or column a primary unsafe singleton and another sensitive cell, the unique contributor of the singleton can disclose the other cell. Extra protection is needed to avoid this disclosure risk. The software is able to consider singleton problems and add required protection when using Hypercube, Modular or Optimal method.

Linked tables Some confidentiality procedures consist in the simultaneous protection of several tables with cells in common. If a secondary suppressed cell appears in more than one table, it needs to be suppressed in all tables of the set where this cell appears. Dealing with linked tables is possible in τ-Argus when using Modular or Hypercube technique.

Table 1 gives a quick summary of the above qualitative comparisons.

	Hypercube	Network	Modular	Optimal
Free	Yes	Yes	No	No
Negative cell values	Handled	Not handled	Handled	Not handled
Singleton issues	Handled	Not handled	Handled	Handled
Linked tables issues	Handled	Not handled	Handled	Not handled

Table 1: Algorithms implemented in τ-Argus: some qualitative aspects.

3 An empirical study

The Network algorithm is a limited way to protect tables because it can only be used with some 2-dimensional tables and is unable to face singleton problems. To make a global comparison (including 3-dimensional tables) of methods to solve cell suppression problems, we have decided to keep our attention on the three other methods described in Section 2. In our study, we aim at quantifying the trade-off between loss of information and time-consumption of the algorithms. The study relies on 144 tables grouped into four categories depending on the number of explanatory variables with a special look at some sets of linked tables. We will then point out the audit problems that may arise when using Hypercube or Modular method.

3.1 Data sets

To make the comparison we use real French data sets. They deal with enterprise statistics, mainly structural business and business demography statistics. Most of underlying data used to construct tables come from administrative sources. Constructed tables answer either national or European regulations and concern enter-

prise births, deaths, high growth innovative enterprises, foreign affiliates resident in France... Explanatory variables are economic activities (European NACE classification with more or less details), geographic indicators, legal status and size ranges. We consider 41 frequency tables and 103 magnitude tables. Response variables are economic indicators (turnover, investment...) or size indicators in terms of employees.

The analysis is made on four groups of tables with one, two or three explanatory variables. 42 bivariate tables protected with the linked tables procedure are also included in the study. Table 2 presents the comparisons and the tested methods in each case. More information about data sets of the study is available in Table 7 (appendix).

	Compared methods	Number of instances
One explanatory variable	Hypercube, Modular, Optimal	11
Two explanatory variables	Hypercube, Modular, Optimal	50
Three explanatory variables	Hypercube, Modular, Optimal	41
Linked tables issues	Hypercube, Modular	42

Table 2: Four groups of tables used in the study.

3.2 Principles

Sensitive cells: definition and protection intervals

Two rules described in Nicolas (2010) are applied to protect French business statistics: a three-unit rule and the dominance (1,85) rule. Let a cell $T_C = \sum_{i=1}^{n} w_i x_i$ where $x_1 \geq \ldots \geq x_n$. x_i denotes the response value and w_i its associated weight. The cell is considered sensitive if $\sum_{i=1}^{n} w_i < 3$ (frequency rule) or $x_1 > 0.85 \times T_C$ (dominance rule).

For the three methods tested, additional protection against singleton disclosure is provided.

Additivity and non-negativity constraints of the published table enable an intruder to derive estimates from an unpublished cell. τ-Argus computes protection intervals to ensure that a sensitive cell cannot be estimated too precisely. We consider the following protection intervals:

$$[0.9 \times T_C,\ 1.1 \times T_C] \text{ for a frequency problem.}$$

.

$$\left[2 \times T_c - \frac{x_1}{0.85},\ \frac{x_1}{0.85}\right] \text{ for a dominance problem.}$$

Note that if a cell is sensitive because of the two sensitivity rules, e.g. a cell with a unique contributor, protection intervals are computed as if it is a dominance problem.

The protection is made with respect to protection intervals and the cost function is equal to the cell value for all three methods.

Audit routine

An intruder can compute for each suppressed cell a feasibility interval. She or he can derive upper and lower bounds of this interval by solving a linear programming problem under constraints of non-negativity and additive structure of the table. See Fischetti and Salazar (2000) for more details. The audit routine implemented in τ-Argus computes these bounds for all suppressed cells and points out audit problems, i.e. cells where the protection interval is not included in the feasibility interval. For these cells an attacker is able to compute a precise estimation.

The Optimal procedure does not provide any audit issues because the possible solutions are restricted to a set of feasible suppression patterns with respect to protection intervals of sensitive cells. The Hypercube and Modular techniques can lead to audit problems because the protection of unsafe cells is made at a sub-table (without hierarchical substructure) level. Considering the additivity relations at a table level gives new constraints and the resulting feasibility intervals are smaller. For more details about risk models see Hundepool and al. (2012). The proportion of primary suppressed cells with insufficient protection after secondary cell suppression is computed in the study.

4 Computational results

4.1 Time-consumption

The time allowed for convergence is set to 10 minutes for the Optimal method. After this time the algorithm stops and the suppression pattern proposed by the software is accepted. Optimal patterns taken into account in this Section are either the final result obtained before 10 minutes or the one proposed after the time limit. For the Modular solution the maximum time per sub-table is also fixed to 10 minutes. It appears clearly that table dimension has an impact on time-consumption of algorithms. Hypercube and Modular approaches always find a quick solution for 1- and 2-dimensional tables. For trivariate tables with the Modular technique, there are 10 cases out of 41 where the solution is not given before 10 minutes. However, in 9 of these cases, the table does not include any hierarchical variable: the CSP problem cannot be reduced by using the table structure, Optimal and Modular approaches lead to the same suppression patterns. For the last table, the Modular solution is found after 16 minutes. The Optimal suppression pattern is obtained before the reference time in 10 instances out of 11 1-dimensional tables, and in 45 bivariate tables out of 50. The Optimal solution is reached for only 10% of trivariate tables. A solution is found in less than 10 minutes for all linked tables of this study when using both Hypercube and Modular methods.

Table 3 gives the average execution times of cell suppression algorithms for the groups of tables considered in this study. The Hypercube implementation converges very quickly: solution is obtained after an average time of 2, 1 and 4 seconds for 1-, 2- and 3-dimensional tables, respectively. Modular solutions are obtained after 7, 7 and 179 seconds for same tables. The Fischetti and Salazar global optimization is the slowest procedure: a solution is found or accepted after an average execution time of 56, 89 and 547 seconds depending of the number of explanatory variables. Note that the research of the best suppression pattern is stopped after 600 seconds: distributions of execution times are right-censored. The linked table instances are quickly solved here but the study is limited to simple cell suppression problems with a few sensitive cells.

Explanatory variables	Hypercube	Modular	Optimal
1	2 (2)	7 (17)	56 (180)
2	1 (0.4)	7 (18)	89 (196)
3	4 (7)	179 (252)	547 (162)
2 - Linked tables	1 (0.2)	2 (1)	-

Table 3: Average execution times in seconds (standard deviation).

4.2 Loss of information

Two criteria are considered to measure loss of information: sum of values in secondary suppressed cells that corresponds to the cost function, and number of suppressed cells. The free Hypercube method is the reference one. Table 4 shows that using a commercial solver produces better results looking at the cost function. For univariate tables, the average cost is 26% or 31% lower when using Optimal or Modular approach. However there is a non-negligible gain for only 3 (out of 11) tables of the data set that have deep hierarchical structures. Information gain is higher for 2- and 3-dimensional tables: cost of cell suppression with a commercial algorithm is reduced by about 40% and 70%, respectively. Note that the Optimal solution sometimes leads to a more costly suppression pattern. In such instances the solution reached by the Modular implementation is actually not a feasible one: additivity relations added at a table level enable an attacker to derive smaller feasibility intervals for some cells. See Section 4.3 for complementary results about audit problems that arise with the use of a method working on non-hierarchical sub-tables. Information loss with the Hypercube solution for linked tables is about twice than suppression cost with Modular approach.

Number of suppressed cells is also a good indicator of information loss: it is the first obvious measure for a data user. For univariate and bivariate tables there are only slight differences between optimization methods. For 3-dimensional tables tested here the number of suppressed cells increases compared to the Hypercube

method when calling a Modular or Optimal technique. Some structures of suppression hide a lot of small cells, up to 18 and 22 times more cells than the Hypercube suppression pattern. The global optimization procedure tends to suppress more cells than the Modular approach because more elaborated suppression patterns can be found.

Explanatory variables	Hypercube		Modular		Optimal	
	Cells	Cost	Cells	Cost	Cells	Cost
1	1.0 (0.0)	1.0 (0.0)	0.90 (0.23)	0.69 (0.39)	1.40 (1.42)	0.74 (0.39)
2	1.0 (0.0)	1.0 (0.0)	0.90 (0.20)	0.59 (0.30)	0.94 (0.30)	0.56 (0.26)
3	1.0 (0.0)	1.0 (0.0)	2.40 (3.79)	0.31 (0.16)	3.09 (3.91)	0.32 (0.15)
2 - Linked	1.0 (0.0)	1.0 (0.0)	0.88 (0.23)	0.53 (0.26)	-	-

Table 4: Average information loss in terms of number of cells and cell cost (standard deviation). The hypercube method is the reference one.

4.3 Audit issues

After computation of suppression patterns, the audit routine is launched to detect cells with insufficient protection, i.e. cells where the required protection interval defined in Section 3.2 is not included in the feasibility interval an intruder can derive. Table 5 presents proportions of primary suppressed cells with an audit problem. All protection intervals are respected with the Fischetti and Salazar methodology. This is an empirical prove that the implementation of the algorithm in τ-Argus is efficient.

However it remains some disclosure risk when using a method working with sub-tables without hierarchical structure. The sub-table risk model used to ensure protection against disclosure by differencing does not take into account additivity relations between sub-tables. The results about univariate instances should be interpreted with extra care given that only 2 out of 11 tables are concerned by audit issues. For other tables results are somehow different. The Modular approach performs better for 2-dimensional tables and vice versa for tables with 3 dimensions. More than half of trivariate tables protected with the Modular heuristic face at least one audit problem. Only 22% of these tables are concerned when using Hypercube method.

Audit problems arising with linked tables are close for both Hypercube and Modular methods.

Explanatory variables	Hypercube	Modular	Optimal
1	4.2 (10.6)	5.9 (15.7)	0.0 (0.0)
2	2.2 (7.6)	1.7 (7.2)	0.0 (0.0)
3	0.3 (0.6)	1.3 (2.0)	0.0 (0.0)
2 - Linked tables	1.3 (3.7)	1.1 (3.4)	-

Table 5: Average proportions of sensitive cells with insufficient protection (standard deviation) in percentage.

5 Discussion and further developments

This section aims at providing some guidelines about the choice of a method to solve cell suppression problems.

This paper shows that the choice of a secondary cell suppression method consists in a trade-off between information loss, time-consumption and disclosure risk. As outlined in Giessing (2004), techniques based on MILP clearly outperforms the Hypercube heuristic when looking at the suppression cost. The Modular and Optimal techniques generally give quite similar suppression patterns. The Optimal algorithm tends to suppress more cells in some cases, especially for higher-dimensional tables. Hypercube treatments seem to be optimal for univariate tables except for deep hierarchical explanatory variables. For tables with more than 2 dimensions, if the NSI owns a license for a commercial solver, it is better to use a commercial method. Optimal method has the advantage of addressing all disclosure risk scenarios because the underlying risk model is at a table level. All solutions given by the algorithm are feasible and all sensitive cells are sufficiently protected, even if the algorithm does not converge to a unique suppression pattern. Table 6 presents some results for a subset of 26 3-dimensional tables. For these tables the Modular unique solution is reached before 10 minutes but the Optimal suppression pattern is not. Mean number of suppressed cells and their mean cost is compared to the reference Hypercube method. We also give average execution times and the mean proportion of audit issues. The Optimal method appears as the best compromise between avoidance of disclosure risk and size of secondary suppressed cells. The Modular approach and its non-global associated risk model lead to 1.9% of cells with insufficient protection.

Criterion	Hypercube	Modular	Optimal
Execution time (seconds)	2 (0.6)	40 (65)	600 (0.0)
Number of cells	1.0 (0.0)	2.80 (4.51)	3.41 (5.36)
Cell cost	1.0 (0.0)	0.38 (0.10)	0.38 (0.10)
Audit issues (% of sensitive cells)	0.4 (0.7)	1.9 (2.2)	0.0 (0.0)

Table 6: Study results for trivariate tables when the Optimal method does not converge to a unique solution before 10 minutes (standard deviation).

However the number of suppressed cells is sometimes higher with the Optimal approach. Defining an alternative cost function is an alternative to consider both number of suppressed cells and cell sizes. See Giessing (2004) for some examples. In all cell suppression problems, an audit control should be done after table protection

to ensure that disclosure risk is completely avoided.

For linked tables, the disclosure risk induced by the two available methods can't be eliminated directly. A 2-step alternative is proposed in Giessing (2004). After table protection and identification of insufficiently protected cells, the Optimal method is applied to the resulting table to give extra protection for those cells. However, this technique must be applied separately for each table of the set. A checking on new secondary suppressed cells is necessary to ensure that it does not include cells that appear in several tables. This checking is normally quick if the proportion of audit issues is low.

In this paper, the trade-off between time-consumption and information loss of secondary cell suppression methods is analyzed. The Hypercube method is a quick and free heuristic that should be used if a data provider cannot invest money to get a solver license. If a commercial solver is available it is preferable to consider a more elaborate model. Both Modular and Optimal approaches give good results, but the Modular method is not able to eliminate all inferential disclosure risks because of its underlying risk model. Given that the two methods are able to address singleton disclosure in the last release of τ-Argus, it seems that using Optimal algorithm is the best way to solve cell suppression problems. It namely faces all disclosure risks and the research for the best suppression pattern can be stopped after a predefined time limit, without subsequent additional suppression. For high-dimensional tables the method is generally not able to find a unique suppression pattern within a reasonable time. Picking the feasible suppression structure obtained after a short time of research (10 minutes in our study) gives comparable information loss than with the Modular approach.

Today the R package sdcTable is able to use free solvers as glpk. It has not been tested in this study. However Meindl and Templ (2012) show in an empirical comparison that commercial solvers generally outperform the free ones. The future evolution of τ-Argus will enable to call free solvers and solve MILP issues. De Wolf (2012) indicates that the first open-source release of τ-Argus with implementation of free solvers is planned for the end of 2014.

References

Castro, J. (2002). Network Flow Heuristics for Complementary Cell Suppression: An Empirical Evaluation and Extensions. In: *Inference Control in Statistical Databases*, Springer (Lecture Notes in Computer Science), **2316**, 59-73.

De Wolf, P.P. (2002). HiTaS: a Heuristic Approach to Cell Suppression in Hierarchical Tables. In: *Inference Control in Statstical Databases*, Springer (Lecture Notes in Computer Science), **2316**, 74-82.

De Wolf, P.P. (2012). Argus open source. Presentation for the fourth meeting of the Expert Group on Statistical Disclosure Control (Luxembourg, 11-12

December 2012).

Fischetti, M. and Salazar, J.J. (2000). Models and Algorithms for Optimizing Cell Suppression Problem in Tabular Data with Linear Constraints. In: *Journal of the American Statistical Association*, **95**, 916-928.

Giessing, S. (2004). Survey on Methods for Tabular Data Protection in ARGUS. In: *Privacy in Statistical Databases*, Springer (Lecture Notes in Computer Science), **3050**, 1-13.

Giessing, S. and Repsilber, D. (2002). Tools and Strategies to Protect Multiple Tables with the GHQUAR Cell Suppression Engine**.** In: *Inference Control in Statistical Databases*, Springer (Lecture Notes in Computer Science), **2316**, 181-192.

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Schulte Nordholt, E., Spicer, K. and de Wolf, P.P. (2012). Statistical disclosure control. *Wiley Series in Survey Methodology.*

Hundepool, A., van de Wetering, A., Ramaswamy, R., de Wolf, P.P., Giessing, S., Fischetti, M., Salazar, J.J., Castro, J. and Lowthian, P. (2011). *τ-Argus user's manual, version 3.5.*

Meindl, B. and Templ, M (2012). Analysis of commercial and free and open source solvers for linear programming. Deliverable from the Essnet Project on Common Tools and Harmonized Methodologies for SDC in the ESS.

Nicolas, J. (2010). La gestion du secret dans les tableaux diffusant des statistiques d'entreprises. In: *La Lettre du SSE*, **65** (French working paper).

Appendix

Explanatory variables	Total cells Mean (sd)	Empty cells Mean (sd)	Primary suppressed cells Mean (sd)
1	481 (454)	30 (94)	24 (22)
2	1607 (2913)	623 (2306)	102 (274)
3	3411 (1438)	696 (729)	459 (389)
2 - Linked tables	859 (191)	19 (20)	33 (20)

Table 7: Structure of tables used for comparison.
Note: One specific table with 9984 sensitive cells out of 14655 is not considered to compute the structure of univariate tables.