CASC PROJECT

Computational Aspects of Statistical Confidentiality

December 2003

Test report on implementation of cell-suppression algorithms based on linear programming techniques in τ-ARGUS 2.2

Autor: Sarah Giessing

Institute: Federal Statistical Office of Germany

Deliverable No: 6-D7

11.12.2003

Test report on implementation of cell-suppression algorithms based on linear programming techniques in τ-ARGUS 2.2

Deliverable 6-D7

Sarah GIESSING,
Federal Statistical Office of Germany
65180 Wiesbaden
E-mail: sarah.giessing@statistik-bund.de

1. *Introduction*

τ-ARGUS is a software package for tabular data protection, offering to protect tables by cell suppression. The process of cell suppression involves mainly two steps: The set-up of the table with the identification of sensitive cells that might reveal individual information, if they were published. τ-ARGUS offers to identify, and suppress those cells. In the second step, in order to prevent these so called "primary suppressions", or "sensitive" cells, from exact disclosure, or from being closely estimable from the additive relationship between the cells of the table, additional cells (so called "secondary" or "complementary" suppressions) must be suppressed. This second step is called "secondary cell suppression".
The problem of finding an optimum set of suppressions is known as the 'secondary cell suppression problem'. It is computationally extremely hard to find exact, or close-to-optimum solutions for the secondary cell suppression problem for large hierarchical tables. τ-ARGUS offers a variety of algorithms to find a valid suppression pattern even for sets of large hierarchical tables linked by linear interrelations. It is up to the user to trade-off quality vs. quantity, that is to decide how much resources (computation time, costs for extra software etc.) he wants to spend in order to improve the quality of the output tables with respect to information loss. The package offers a choice between different approaches:

OPTIMAL Fischetti/Salazar methodology aims at the optimal solution of the CSP [4]. A feasible solution is offered at an early stage of processing, which is then optimized successively. It is up to the user to stop execution before the optimal solution has been found, and accept the solution reached so far. The user can also choose the objective of optimization, i.e. choose between different measures of information loss. Note that the method relies on high performance, commercial OR solvers. It should also be mentioned here that some problems have not yet been fully solved in the current version of τ-ARGUS: It may happen that the algorithm considers a single respondent cell to be properly protected even though there is only one other suppression in the same row/column/... of the table and this suppression is another single respondent cell (see also method SINGLETON below).

MODULAR The HiTaS method [3] subdivides hierarchical tables into sets of linked, unstructured tables. The CSP problem is solved for each subtable using Fischetti/Salazar methodology [4]. Backtracking of subtables avoids consistency problems when cells belonging to more than one subtable are selected as secondary suppressions.

NETWORK The concept of an algorithm based on network flow methodology has been outlined in [1]. Castro's algorithm aims at a heuristic solution of the CSP for two-dimensional tables. Network flow heuristics are known to be highly efficient. It may thus turn out that the method is able to produce high quality solutions for large tables very quickly. τ-ARGUS offers an implementation applicable to 2-dimensional tables with hierarchical substructure in one dimension. 'Network' is offered in two

alternative variants. A license for a commercial OR solver is not be required to run the algorithm. Two free solvers are supplied instead.
In the current implementation, the protection of single respondent cells (see above) has not yet been solved.

HYPERCUBE The hypercube algorithm GHM*ITER* developed by R.D. Repsilber (see [5,8]) is a fast alternative to the above three OR based methods. This heuristic is able to provide a feasible solution even for extremely large, complex tables without consuming much computer resources. The user, however, has to put up with a certain tendency for over-suppression.

SINGLETON Special application of GHM*ITER*, addressing only the protection of single respondent cells. The method is meant to be used as preprocessing for the OPTIMAL and NETWORK methods, for which a solution for the problem with single respondent cells mentioned above has not yet been implemented.

With respect to GHM*ITER* and the modular method, both involving backtracking of subtables, it should be noted that such methods are not '*global'*. This causes a certain disclosure risk (see [2] for problems related to non-global methods for secondary cell suppression.)

This document provides a report on the results of tests carried out to confirm particularly the correct implementation of the Linear Programming based algorithms 'Optimal', 'Modular', and 'Network'. In section 2 below, we briefly explain the test scenario, and present the results of the tests. Section 3 summarises the results.

2. *Empirical Tests*

For the testing, we used a variety of hierarchical tables generated from the synthetic micro-data set supplied as CASC deliverable 3-D3. In particular, we generated 2 and 3-dimensional tables, where one of the dimensions had a hierarchical structure. Manipulation of the depth of this hierarchy resulted in several different, hierarchical tables (tables no 1 to 5 in table 1 below). As the implementation of the network flow method available at the time of testing was not applicable to hierarchical tables yet, some non-hierarchical subtables of tables 1, 2, and 3 were added to the set of tables (tables 1a, 1b, 2a, and 3a). Methods 'Optimal' and 'Modular' were applied to the hierarchical tables, using CPLEX 7.5 as OR-solver. All possible combinations of each variant of the 'Network' method with any of the free OR-solvers (for details see [6, sec. 2.10]) were applied to then non-hierarchical tables 1a, and 1b, 2a, and 3a. .

Table 1. Structure of test tables

		Number of Cells		
Table	**Hierarchical levels**	**Total**	**Zero**	**Primary Suppressions**
		2-dimensional tables		
1	3	460	71	18
1a	-	170	29	5
1b	-	310	53	13
2	4	1050	168	60
2a	-	610	108	43
3	6	8230	2061	989
3a	-	5010	1396	663
		3-dimensional tables		
4	3	8280	2740	848
5	4	18900	6937	2198

Basically, all algorithms are expected to be self auditing, that is, they are supposed to determine secondary suppressions ensuring that the resulting feasibility interval of any sensitive cell exceeds upper and lower protection levels computed by τ-ARGUS. Nevertheless, non-globality of methods 'Modular' and 'Hypercube' is a possible source of disclosure risk. Table 2 gives empirical confirmation of this problem. It shall be stressed, however, that this is not evidence of incorrect implementation, but due to properties (i.e. non-globality) of the algorithms.

The algorithm to compute the feasibility intervals has been implemented at University Ilmenau ([7], deliverable 3 D-4), and tested using an implementation of the problem developed by Statistics Netherlands. It was then applied to the output of secondary cell suppression. For each primary suppression of each output table, we checked whether the feasibility intervals satisfy the conditions imposed by the protection levels. In none of the tables we found insufficient protection in a suppression pattern computed by 'Optimal', or 'Singleton/Optimal'. This gives some empirical prove of the mathematical and software reliability of 'Optimal' also underlying the 'Modular' method. Neither did we find insufficient protection in a suppression pattern computed by any variant of 'Network' for tables 1a, 1b, and 2a. For the largest of the non-hierarchical tables, one variant of 'Network' failed (combination '0-1 flows'/PPRN-solver, see [6]). The other two variants finished successfully, leaving, however, two of the cells slightly underprotected. As in all these cases protection levels very low (2 units), and cells were underprotected by just 1 unit, this seems rather be due to problems of numerical precision than to incorrect implementation of the algorithms.

Table 2. Audit results: Percentage of primary suppressions with insufficient protection

Table	# Dim	Hierarchical levels	Primary Suppressions	Percentage of primary suppressions with insufficient protection	
				Mod	Hyp
1		3	18	5.56	0.00
2	2	4	60	5.00	0.00
3		6	989	5.46	1.31
4	3	3	848	2.59	0.24
5		4	2198	2.14	5.64

3. *Summary*

In order to empirically prove the reliability of the linear programming based secondary cell suppression algorithms in τ-ARGUS, we ran extensive tests of the methods 'Optimal', 'Singleton/Optimal', and 'Network'. Results were encouraging: In no case any of these algorithms produced a suppression pattern failing severely to meet the protection requirements imposed on the sensitive cells. As algorithm 'Optimal' is also underlying the method 'Modular', this is also prove of reliability of this method. It must be noted however, that both methods, 'Modular', and 'Hypercube' actually do often generate a suppression pattern not meeting the safety requirements of some primary suppressions. But this is due to methodological issues of non-global methods – not to an incorrect implementation in τ–ARGUS.
It is important to note that the current version of τ-ARGUS does not provide a specific solution for methods 'Optimal' and 'Network' with respect to single respondent cells: It may thus happen that these algorithms consider a single respondent cell to be properly protected even though there is only one other suppression in the same row/column/... of the table and this suppression is another single respondent cell. These respondents would then be able to disclose each others contribution. To solve this problem, τ-ARGUS offers the pre-processing method SINGLETON, which should always be used in advance of running 'Optimal', or 'Network' to prevent this kind of disclosure risk.

References

1. Castro, J. (2002), 'Network Flows Heuristics for Complementary Cell Suppression: An Empirical Evaluation and Extensions', In: '*Inference Control in Statistical Databases*' Domingo-Ferrer (Ed.), Springer (Lecture notes in computer science; Vol. 2316)
2. Cox, L. (2001), 'Disclosure Risk for Tabular Economic Data', In: *'Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies'* Doyle, Lane, Theeuwes, Zayatz (Eds), North-Holland
3. De Wolf, P.P. (2002), 'HiTaS: A Heustic Approach to Cell Suppression in Hierarchical Tables', In: '*Inference Control in Statistical Databases*' Domingo-Ferrer (Ed.), Springer (Lecture notes in computer science; Vol. 2316)
4. Fischetti, M, Salazar Gonzales, J.J. (2000), 'Models and Algorithms for Optimizing Cell Suppression Problem in Tabular Data with Linear Constraints', in *Journal of the American Statistical Association*, Vol. 95, pp 916

5. Giessing, S., Repsilber, D. (2002), 'Tools and Strategies to Protect Multiple Tables with the GHQUAR Cell Suppression Engine', In: '*Inference Control in Statistical Databases*' Domingo-Ferrer (Ed.), Springer (Lecture notes in computer science; Vol. 2316)
6. Hundepool, A., van de Wetering, A., de Wolf, P.P., Giessing, S., Fischetti, M., Salazar, J.J., Caprara, A. (2002), τ-ARGUS *users's manual, version 2.1*
7. Rabenhorst, A. (2003), 'Bestimmung von Intervallen und Ersatzwerten für gesperrte Zellen in statistischen Tabellen', Diploma Thesis, Manuscript, University Ilmenau.
8. Repsilber, D. (2002), 'Sicherung persönlicher Angaben in Tabellendaten' - in *Statistische Analysen und Studien Nordrhein-Westfalen, Landesamt für Datenverarbeitung und Statistik NRW,* Ausgabe 1/2002 (in German)