CASC PROJECT

Computational Aspects of Statistical Confidentiality
December 2002

Test report on implementation of GHQUAR in τ-ARGUS 2.1

Sarah GIESSING,
Federal Statistical Office of Germany
65180 Wiesbaden

Deliverable 6-D3

31.12.2002

Test report on implementation of GHQUAR in τ-ARGUS 2.1

Deliverable 6-D3

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.
The GHQUAR algorithm developed at the Statistical office of Northrhine-Westphalia/Germany offers a quick heuristic solution. It is one of the tasks of CASC project to extend the software τ-ARGUS into a control centre for tabular-data protection, particularly making GHQUAR easily available through τ-ARGUS. Meanwhile, a new version of the program, GHMITER, has replaced the GHQUAR software. Therefore we refer to GHMITER instead of GHQUAR in the following.

This document provides a report on the implementation of GHMITER in version 2.1 of τ-ARGUS. In 1.1 below, we briefly explain the test scenario. In section 2 we describe technical problems of the GHMITER implementation found during the testing and suggest corrections, or improvement. Sections 3 provides suggestions for general innovation of the interface in upcoming versions, with a particular view to problems stated by Eurostat in section 3.1.

1.1. Test data

We tested the GHMITER implementation with a variety of tables generated from the synthetic micro-data set supplied for the CASC deliverable 3-D3. In particular, we generated 3-dimensional tables, where one of the dimensions had a hierarchical structure. Manipulation of the depth of this hierarchy resulted in basically four different tables, with a total number of cells varying between 8.2 tsd. and 150 tsd. cells. Secondary cell suppression was applied to each table, using both, the hypercube method (GHMITER engine), and the HiTaS method with CPLEX as LP-solver. Except for the HiTaS run on the big 150 tsd. cell instance, all runs were completed successful. A CD-ROM with the results of the runs was produced and sent to CASC partners for comparison, and to the project management.

2. *Technical problems of the implementation*

In order to run GHMITER on a single table, two control files, STEUER and TABELLE must be supplied, together with the datafile EINGABE. The interface τ-ARGUS extracts the information required from the user input, sets up the table, and creates those files.
We found the implementation basically successful: It is possible to run GHMITER properly using τ-ARGUS as interface. However, both programs must be improved to come to a more stable implementation.

Section 2.1 lists problems concerning the interface, while section 2.2 relates to issues concerning the GHMITER engine.

2.1. Problems caused by the τ-ARGUS interface

File TABELLE

1st parameter in line 3: Should be set to 0.000005

2nd parameter in line 3 MAXW:

- There should be a '.' instead of a ',' used as decimal-separator,
- The parameter is used for the internal logarithmic scaling of the cell values, and has therefore some impact on the selection of secondary suppressions. It should exceed the overall total of the table (of all tables in a table-to-table protection run), while still being as close as possible to this overall total. Problems may occur for instance, when MAXW is presented in floating point format, while the datafile stores huge values (such as the overall total) in exponential format, e.g. contains (up-) rounded values.

File STEUER

- The range ratio parameter is passed over to file STEUER with a precision of 4 digits only: as a result no runs with range ratio < 0.01 can be performed! In certain situations users might wish to use a ratio, which is not zero, but very close to it. This will on one hand provide the week protection corresponding to a minimum frequency rule, while on the other hand still prevent 'frozen' cells to be selected as secondary suppressions. The ratio parameter should either be passed with a precision of 16 digits, or the interface should explicitly offer the choice of an 'infinitely' small, nonzero ratio, which should then be set to be smaller as 1/MAXW by the interface.
- The *q*-parameter of the (*p,q*)-rule is stored internally in 2 digits only, which will in certain situations lead to the problem that – even though the user wanted it to be 100 – the programs assume a *q*-parameter of 10. Consequently, the GHMITER run tends to fail, because it is usually impossible to protect a table with this method when 10% *a priori* bounds are assumed for all cells in the table.
- Codes given by the user for missing values of explanatory variables lead to inappropriate sorting of the input file, causing breakdown of GHMITER, when they are shorter as the length of the variable. The interface should prevent the user from such misspecification.
- There is an inconsistency in the definition of the protection levels and *a priori* bounds between the implementations of GHMITER and HiTaS:
  - In the GHMITER implementation, *a priori* bounds depend on the *q*-parameter of the (*p,q*)-rule, whereas in the HiTaS implementation they are fixed to 50% of the cell value. The bounds of the HiTaS implementation should also depend on a user defined *a priori* bounds parameter.
  - In the HiTaS implementation, the ratio *p/q* is used to determine the protection level, whereas in the GHMITER implementation the protection level ('ratio parameter') depends on the parameter *p* only. The later setting was proposed assuming that users, although they may want to change *a priori* bounds, they may not want to (and not be aware of the effect) change the safety level.

  For upcoming versions it is suggested that the interface enables the user to change the *a priori* bounds (*q*-parameter), while still keeping the original ratio $R := p/q$ used to determine the safety level. In such an implementation the GHMITER ratio parameter should be determined as $2R$ instead of $2\frac{q}{100}R$.

  This modification could be achieved, for instance, by asking the user for the parameter *p* of the *p*%-rule only, instead of the two parameters *p* and *q* of the (*p,q*)-rule, and by offering him specification of an additional *a priori* bounds parameter (which should then not be referred to as '*q*'). The parameter *q* would be assumed to be 100.

2.2. Problems caused by the GHMITER engine

GHMITER checks the feasibility of each hypercube, one by one. The method is unable to 'add' the protection given by multiple hypercubes. In certain situations, particularly when rather tight *a priori* bounds have been defined by the user, it is not possible to protect a particular sensitive cell by suppression of one single hypercube. In such a case, GHMITER is unable to confirm that this cell has been protected properly, according to the safety level given by the ratio parameter. The current implementation thus requires the user to find out, in a time consuming trial and error process, which choice of the ratio parameter leads to a successful run, causing him to reduce the safety level for *all* sensitive cells, even though it may have been only *a few*, the protection of which could not be confirmed. Note that use of GHMITER with symmetric *a priori* bounds of 100 % and less is quite novel, and has not been tested in practice before the implementation into ARGUS.
Meanwhile test results are available. They exhibit that on one hand the implementation with fixed symmetric *a priori* bounds, which allows to reduce the ratio parameter for instance in the case of a *p*%-rule from 1+*p* to 2**p* (see [1]), leads to an considerable improvement of the suppression pattern, while still providing the same amount of safety to the primary suppressions. On the other hand, the assumption of these *a priori* bounds is responsible for low stability of the program, experienced by the ARGUS user.
We therefore suggest to implement a new version of GHMITER into the next version of τ-ARGUS, able to reduce the protection level automatically, and individually, step by step, for those cells, the protection of which the program cannot confirm according to the original user specified protection level (under the *a priori* bounds assumption specified by the user). Although it is impossible for GHMITER to trace the individual cells concerned, the new implementation should offer statistics on the number of those cells, and the amount of protection confirmed.

3. Further suggestions for the τ-ARGUS interface

In the first paragraph 3.1 we suggest some minor improvement and more relevant innovation for the interface program, and consider particular needs of Eurostat in 3.2.

3.1. General recommendations

(1) Although most testers were generally happy with the **report file**, it might still be improved: We suggest to include summary statistics on the information loss concerning all potential cost variables, not only frequency, such as total value suppressed, or total number of contributions suppressed.
(2) For eventual further processing of the data by the user, it might be useful to offer an **output format** presenting the cell values also for suppressed cells, with an additional entry 'suppression status'. As the upcoming version will be able to read external tables, it should in particular be able to read at least one τ-ARGUS output format.
(3) For some users (in particular Eurostat and German Federal Statistical Office) it might be extremely helpful, if new versions of ARGUS would offer a facility to **'add' external tables** with an identical structure (same explanatory and response variables) but from different sources (e.g. countries) into one new table with one additional explanatory variable for the relation 'total = source 1 + source 2 + ... + source *m*'. Such a facility must be able in particular to contrive the n largest contributions for a 'total' cell of the new explanatory variable, given the input information of the n largest contributions of the corresponding cell in the *m* input files. For simplicity, the case that a contributor to a cell in source *i* might be identical to a contributor to the same cell in source *j* should be ignored. If information on such cases is available, it will in principle still be possible for the user to correct the cell information manually.
(4) For some users it would be extremely useful, when ARGUS offers a facility to **split external hierarchical tables** into a set of separate subtables, to be protected separately. A typical example are tables on commuters: the number of commuters between communities (of a region) does not

add up to what is published on the next level of the hierarchy, namely the number of commuters between regions (the regional total of the commuters between the communities of a region will neither be computed nor be published). Another example is the product classification, which is on one hand a typical hierarchical classification, but on the other hand, totals on some levels are not published.
In such cases, cell suppression would have to be performed separately on multiple subtables (which may be hundreds, or even thousands). A simple option for a user to define which levels of a hierarchy should *not* be considered as creating links between subtables would improve the usability enormously in such a case. Given this information, ARGUS should then split the input table, generate the independent subtables, and do the secondary suppression for each one of them separately – all this on a single 'command' of the user.
Another application could be the case of multiple response variables the user wants to be protected with each variable as its own shadow variable. The user might then create a 'pseudo' relationship between those variables (a hierarchy consisting only of the total level, and the level of the individual variables). He would then indicate that he wants the relation to this (artificial) total to be ignored and have ARGUS protect the different tables (with different response variables) separately within a 'single run'.

3.2. SBS requirements for τ-ARGUS

In the document 'ANNEX to note ESTAT/D-2/PF/mhd/20300 of 12 December 2002' Eurostat has explained some specific requirements of the users in the Eurostat SBS section. In the following, we comment on the requested facilities except for the issues raised in item 13. of this document. With respect to these issues we refer to the facilities suggested in 3.1 (4) above as an eventual approach to solve the problem.

(1) Concerning the issue of **non-confidential singletons because of 'wavers'** given to the statistical offices (c.f. item 8. of the Eurostat document), we suggest for the GHMITER method, that τ-ARGUS changes the number of respondents entry in the GHMITER input file from 1 into 3, and uses the option to 'freeze' these cells, thus making then unavailable for secondary suppression.
(2) The issue raised in item 10. of the Eurostat document is the issue of defining the **protection level** for a sensitive cell according to the cell sensitivity, which leads to small protection requirements for cells that are only slightly sensitive. In τ-ARGUS, all algorithms for secondary cell suppression will define the protection levels in this way, except for GHMITER. For GHMITER, it would require a major modification of the program code to change the definition of the protection interval. There is currently no capacity available (in terms of manpower) to do this.
(3) In item 16., Eurostat complains that in table-to-table protection procedures GHMITER does not accept input files, when cells which should be identical according to their definition, differ slightly in their cell value, or number of contributors. The next version of τ-ARGUS will offer table-to-table protection only with GHMITER as engine for the secondary cell suppression. The problem mentioned by Eurostat can be solved by some pre-processing of the tables. Such a pre-processing procedure should detect those **slight discrepancies**, and change cell values in one file in such a way that they match corresponding values in the other. Eventually a post-processing is required to undo this modification after the secondary suppression process.

4. *Summary*

Extensive tests of the GHMITER implementation in τ-ARGUS were performed, with encouraging results: It is possible to run GHMITER properly using τ-ARGUS 2.1 as interface. However, both programs must be improved to come to a more stable implementation. This paper provided several suggestions for improvement and correction of the current version. Apart from this, we also propose innovation for upcoming versions of the program, with a particular view on requirements of Eurostat as expressed in document 'ANNEX to note ESTAT/D-2/PF/mhd/20300 of 12 December 2002' .

References

[1] 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)