STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Joint ECE/Eurostat Work Session on
Statistical Data Confidentiality
(Thessaloniki, Greece, 8-10 March 1999)

COMMISSION OF THE
EUROPEAN COMMUNITIES

EUROSTAT

Working Paper No. 12
English only

Topic (ii): software and computing developments

A SURVEY ON SOFTWARE PACKAGES FOR AUTOMATED SECONDARY CELL SUPPRESSION

Submitted by the Federal Statistical Office of Germany[1]

Contributed paper

1. The paper presents brief descriptions of various cell suppression software packages, focussing on availability, costs, platforms, and the underlying methodology. The systems performances will be compared with respect to information loss, and computing time requirement; other key qualities of the software will be discussed.

I. INTRODUCTION

2. When cell suppression is used as a Statistical Disclosure Control Technique, table cells are suppressed, if the data disseminator considers them revealing too much. To prevent these so-called "primary suppressions", or "sensitive" cells from exact disclosure or a too narrow estimation from the additive relationship between the cells of the table, additional cells must be suppressed.

3. The "Secondary Cell Suppression Problem" is to apply these complementary suppressions to the set of sensitive cells, in such a way as to ensure that the complementary suppressions:

- create the required uncertainty about the true values of the sensitive cells while still
- preserving as much information in the table as possible.

The Secondary Cell Suppression Problem can be stated mathematically as (Integer)-Linear Programming ((I)LP) problem. However, solving those large LP-problems for real-life sized statistical tables is far from easy.

[1] Prepared by Sarah Giessing.

I.1 Systems Included in the Comparison

4. The survey is on five existing software packages for table protection by cell suppression, four of them already in regular use, and one prototype.

Table 1: Systems Included in the Comparison

Name of software	**Software development**[1]	**Platforms / Programming Languages / availability / costs**
GHQUAR	Landesamt für Datenverarbeitung und Statistik Nordrhein-Westfalen (Dietz Repsilber)	FORTRAN codes for IBM and SIEMENS mainframes, non-commercial system
USBCSUP	US-Bureau of the Census (Bob Jewett)	FORTRAN code for DEC computer, non-commercial system
CONFID	Statistics Canada (Gordon Sande / Dale Robertson)	FORTRAN (RATFOR) code for IBM mainframe, SUN SPARC workstation, non-commercial system
ACSSuprs	Sande and Associates, Inc. (Gordon Sande)	Improved version of CONFID, commercial system
***τ-ARGUS* Version 1.5**	CBS Netherlands (second prototype version)	C++ code, WINDOWS-software (32 BIT), non-commercial system, available for free at Statistics Netherlands, additional commercial software required (at ca. 1000 - 2000 EUR)

[1] For names and addresses of contact persons, and further information relating to hardware requirements, availability, and portability aspects see appendix 1.

I.2. Methodology

5. There is broad variety in the methodology applied:

- a very sophisticated algorithm for exact solution of the ILP-problem in τ-ARGUS 1.5,
- a heuristic LP-relaxation approach in CONFID and ACSSuprs,
- iterative procedures in USBCSUP and GHQUAR, which subdivide complex structured tables into subtables, and consider as feasible solutions suppression patterns of a certain structure only, thereby reducing the enormous computational burden effectively.

For brief methodological descriptions see appendix 2.

6. Considering the underlying methodology, we would expect, that in certain situations some programs will perform better than others. How much such differences matter in practice is another question.

II. COMPARISON OF SOFTWARE PERFORMANCE ON REAL LIFE TABLES

7. For the comparison of secondary cell suppression software we chose seven two- and three-dimensional real-life tables, some presenting data relating to turnover, and some presenting data regarding numbers of employees. The most complex set of row relations resulted from an elaborate breakdown of industries (630 rows within a hierarchical system of six levels, created by submarginals

within the classification). See [7] for detailed descriptions of table structures, definitions of upper tolerances and further information on the experiment. On these tables, runs of GHQUAR were conducted at the "Landesamt für Datenverarbeitung und Statistik Nordrhein-Westfalen" on an IBM mainframe. The US Bureau of the Census kindly supplied USBCSUP which was run on the SIEMENS mainframe at the Federal Statistical Office in Wiesbaden.

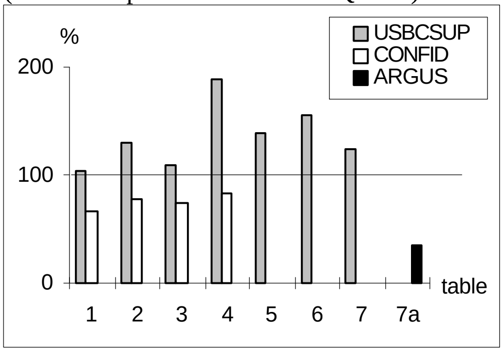
8. During a visit to Statistics Canada, CONFID runs on the two-dimensional tables were performed successfully on a SUN workstation[1] . However, due to the time restriction of only one week for the visit, as well as massive computation times of the program on large tables, unfortunately CONFID runs on 3D-tables could not be completed. As the suppression algorithm is the same for CONFID and ACSSuprs, leading to very similar results on two and three dimensional tables, ACS runs were not performed.

9. Due to their structure[2] τ-ARGUS could not represent any of the original tables directly. We therefore ran τ-ARGUS, and GHQUAR either, on a particular subtable (the "total" column) of the original 3D table 7. Handling the submarginals of the resulting 2D table 7a as additional dimension, we could define table 7a to τ-ARGUS as a 3D table, actually consisting of zero-valued cells prevailingly. In addition to that, we tried to use τ-ARGUS for protection of the original table 7 using a rather plain and simple method: We applied τ-ARGUS seperately to any of the 10 table 7 subtables, each presenting one of the different table 7 response variables, the marginal of which is actually identical to the response variable of table 7a, and their submarginals identical to the response variable of one of the other subtables. Then we suppressed every cell in either subtable, in case it had been suppressed in the actual one, or in one of the other subtables.

II.1. Results of the Experiment

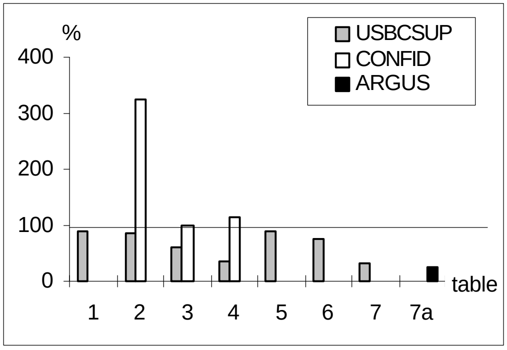
10. Results were rather similar for the seven original test tables. Figures 1 and 2 show for all tables performances of CONFID, USBCSUP, and τ-ARGUS (on table 7a) relative to that of GHQUAR (except for table no.1, for which the CONFID total value suppressed was not recorded).

Figure 1: Number of Suppressions
(relative to performance of GHQUAR)



(GHQUAR=100%)

Figure 2: Total value suppressed[3]
(relative to performance of GHQUAR)



(GHQUAR=100%)

For more detailed inspection of our results, see [7].

11. On the only (test-) table 7a, which it could be applied to directly, τ-ARGUS performed extraordinarily well, with respect to both: number, and total value of the suppressions. The simple approach reported above for using τ-ARGUS to protect the entire table 7 resulted in poor performance

[1] Statistics Canada confessed, that CONFID "was not in a fashion for outside distribution".
[2] Submarginals in every variable, and decompositions of the response variables, c.f. 2.1, below
[3] Total of the original values of the complementary suppressions

of 150 % more complementary suppressions than with GHQUAR. (The figures do not present this result.) On the original tables, overall, CONFID clearly gave the best performance regarding the number of suppressions, whereas USBCSUP suppressed the smallest total value, a direct result of differences in the software design, regarding the assessment of information loss (cf. 3.2, below). The hypercube-program gave reasonable results with respect to both these criteria. Runs of GHQUAR performed earlier with zero safety range, yielding protection against exact disclosure only, gave much better results with fewer suppressions compared to the CONFID output. This suggests, that performance depends largely on the definition of the safety ranges for the primary suppressions.

Table 3: Computation times required

	size (total non-zero cells)	CPU - time used (hours)			
table		USBCSUP [1)]	CONFID [2)]	GHQUAR [3)]	τ-ARGUS [4)]
1	6302	00:01:05	00:01:45	00:00:04	- [5)]
2	4630	00:01:20	< 5 minutes	00:00:03	- [5)]
3	1735	00:00:02		00:00:01	- [5)]
4	1312	00:01:19		00:00:01	- [5)]
5	47374	01:06:55	- [4)]	00:01:39	- [5)]
6	53045	01:06:55[6)]	- [4)]	00:01:34	- [5)]
7	17040	01:31:44[6)]	> 4 hours [5)]	00:02:34	- [5)]
7a	2295[7)]	- [5)]	- [5)]	00:00:01	00:21:09

[a)] Siemens-mainframe, OSD 3

[b)] SUN Ultra Sparc II workstation, 167 mega Hertz

[c)] IBM-mainframe: IBM 3390, MVS2.

[d)] PC 5/86, WINDOWS-NT 32.

[e)] Runs were not performed/completed.

[f)] Because of substructure present for all three dimensions, the systems own 3D procedure couldn't be applied to tables 6 and 7. They had to be treated as set of interrelated 2D tables, which didn't perform very well with respect to CPU time requirement.

[g)] In the τ-ARGUS representation, table 7a had 15 283 cells (2329 non-zero cells and 12954 zero cells)

12. Though CPU requirements recorded from runs on different machines cannot be compared directly, the differences seem to be quite obvious. The hypercube-algorithm is exceedingly faster than the Linear Programming algorithms of CONFID, and τ-ARGUS, as well as the Network-optimization algorithm of USBCSUP, the latter is however still much faster than both LP-systems CONFID, and τ-ARGUS, which matters especially on 3D-tables.

13. It shall be stressed here, that though certainly there exists some trade-off between computing time requirements and information loss, there does not neccessarily exist a trade-off between the level of data-protection and computing time requirement: The fast hypercube and network optimization systems GHQUAR and USBCSUP are actually capable of producing safe suppression patterns, which ensure the required variability in the values of the suppressed cells.

III. SOFTWARE DESIGN RELATED QUALITIES

III.1. Applicability

14. *ACS/CONFID* : Up to three (ACSSuprs: Seven) dimensional moderate sized[4] tables with any substructure can be handled as a single problem without requiring a partitioning into sub-problems.

τ-*ARGUS*: The present version can handle up to three dimensional tables, without submarginals. Tables presenting decompositions of the response variable cannot be represented as a single table.

GHQUAR can handle up to 7D-tables with submarginals in every dimension.

USBCSUP: The (network)-algorithm in use is restricted to two dimensional problems with tree substructure in one dimension. Using what the Census Bureau calles a "backtracking-facility", three dimensional problems with a tree sub-structure in one dimension, arbitrary sub-structure of the second dimension but no substructure of the third dimension cannot be solved as a single problem, but in an iterative process within a single run.

III.2 Information Loss

15. The following are aspects of information loss:

a) *Assessment of information loss*
The idea of equating a minimum loss of information with the smallest number of suppressions is probably the most natural concept. Yet, experience has shown that this concept often yields suppression patterns with many larger cells suppressed, which seems undesirable, leading to the idea to concentrate on minimising the suppressed total, thus avoiding the suppression of larger cells in favour of many smaller cells, or indeed favour a compromise solution.

Moreover, not only the numeric value, but several other criteria may have an impact on a users perception of a particular cells importance, such as its situation within the table (marginals and submarginals are often rated highly important), or category (certain categories of variables often are considered less important).

b) *Formulation of the required amount of protection*
Using linear programming, table users are able to derive upper and lower bounds for the suppressed cells in a table. All five programs ensure that regarding the primary suppressions the interval given by these lower and upper bounds, the so called “suppression interval” or “interval of uncertainty”, can generally be defined large enough to avoid approximate disclosure.

However, to avoid oversuppression, the suppression interval for any particular sensitive cell should be just wide enough to protect the relevant respondents data against approximate disclosure, and be not wider. Actually, adequate formulation of the required amount of protection in terms of cell sensitivity, instead of cell value,in fact is a means to reduce information loss.

c) *Release of bounds as a means to minimize information loss*
If a table has been protected properly, the above mentioned bounds could be released to the customers, meaning to maximize its information content.

[4] In a floor discussion reported for the Proceedings of the Annual Research Conference of the Bureau of the Census in 1993, Dale Robertson stated, that CONFID has an approximate 20 000 cell limit.

Table 4. Information loss

Software system	Assessment of information loss	Formulation of the required amount of protection	Release of bounds as a means to minimise information loss
ACS/CONFID	A choice is offered between the options to minimize the – number of suppressions, or – total value suppressed, or – total value suppressed of a weight variable varying asymptotically as the logarithm of the cell value, which is indeed a compromise solution between the first two options	Required protection is to be defined according to the "cell sensitivity", which effectively avoids oversuppression.	Bounds are calculated, and can be made available to be published.
τ-*ARGUS*	Minimization of the suppressed total of a user defined weight variable (e.g. the response variable, which is the default option). Though prinicipally cell related, the weight variable is to be defined at micro-data level, which may sometimes be uncomfortable.	Required protection must be determined proportinal to the cell value, which may either lead to oversuppression for large, only slightly sensitive cells, or underprotection for smaller, but strongly dominated primaries.	Bounds are calculated during the suppression procedure, and might be made available (though not in the present version) to be published.
GHQUAR	1st priority: Minimum number of suppressions. 2nd priority: Minimum suppressed total of a weight variable varying asymptotically as the logarithm of the cell value (cf. ACS/CONFID). There is an option to make the program attempt to avoid the suppression of totals and subtotals, or alternatively, to assign weights to the cells: The programm will try to avoid the suppression of cells with large weights, and prefer cells with low weights as complementary suppressions.	Width of the suppression interval to be determined proportionally to the maximum valued corner point of the suppression hypercube. To avoid underprotection in certain cases, this proportion should be chosen sufficiently large. Doing so, causes indeed a tendency for oversuppression.	For each primary suppression, the program computes that potential decrease and increase only, which results from suppression of the corner cells of the hypercube, which was selected for protection of the particular primary suppression. This is in fact merely a lower bound for the protection: suppressions from other hypercubes may add to the variability of the cell. The final upper and lower bounds resulting from all the suppressions in the table are not calculated.
USBCSUP	Minimum of the total value suppressed. There is an option to define cells to be chosen as complements first regardless of their value (usually cells not intended to be published anyway), or to be preferred as complements among cells of similar size.	cf. ACS/CONFID.	Bounds are not available to be published.

III.3 Disclosure Risks

16. All programmes aim at excluding the possibility of exact disclosure or unacceptably narrow estimation of the primary suppressed cells.

a) *Partitioning*

However, there is a risk for the security of the data which results from the process of partitioning, i.e. subdividing a table and protecting the subtables iteratively. Even though each single subtable may be protected sufficiently with suppressions carried over to the other tables, it might still be possible to disclose some suppressed cells, when the entire inter-relationship structure of the table is considered. (See [8] for detailed description and example.) For two dimensional tables the problem is well known as the problem of isolated cells (example in [9]).

b) *Table to table protection in multiple tables*

When tables are linked (i.e. have cells in common) it should be avoided that cells suppressed in one table are published in another, and vice versa. This would require the option to import secondary suppressions from tables processed earlier into the current table and to protect them similarily to primary suppressions. To avoid suppression patterns with isolated cells (see above), ideally, inter-related tables should be treated as a single large, multi-dimensional table with empty regions, corresponding to cells not intended to be published. Due to excessive memory requirements, this approach is not likely to be an option for most real life problems.

Table 5. Disclosure risks

Software system	Partitioning	Table to table protection in multiple tables
ACS/CONFID	If a tables size or the size of a single table, containing all cells from multiple tables is only moderate[4] and table dimensions are three (ACS: seven) or less, then multiple tables can be treated together. Otherwise, suppressions can be carried over from (sub)table to (sub)table.	
τ-ARGUS	Up to four dimensional tables without substructure.	The problem of linked tables cannot be solved properly. The present version generates primary suppressions automatically, according to the suppression rule. Importing secondary suppressions from tables linked to the current table is possible only by labourious manual intervention.
GHQUAR	Up to seven dimensional tables without substructure. A new version, able to treat 7D-tables with hierarchical substructure in every dimension as single problem is in the process of being developed.	There is a program available, which carries over suppressions between related tables, and protects them in an iterative process. A utility-package is in developement, which will offer table-to-table protection for the entire set of tables published from the same survey.
USBCSUP	2D-tables with strictly hierarchical substructure in one of the dimensions only.	Facilities to support table-to-table protection. When different tables use the same column relations, they may be protected within a single run of the program. Generally, to perform table-to-table protection with USBCSUP would require the implementation of utility routines.

c) *Prepublished cells*

To ACS/CONFID, and USBCSUP the user may indicate certain cells as already published, uneligible for suppression. The users of τ-ARGUS, and GHQUAR may assign extra weight

[4] In a floor discussion reported for the Proceedings of the Annual Research Conference of the Bureau of the Census in 1993, Dale Robertson stated, that CONFID has an approximate 20 000 cell limit.

(c.f. 3.2 (1)) to prepublished cells, which will make the programs try not to use them as complementary suppressions.

d) *Frequency data*

With exception of τ-ARGUS, all programs have been designed in the first way for tables presenting magnitude data. When tables present frequency data, security for smaller frequencies may be lacking. For details see [7]. For disclosure control of those tables, secondary cell suppression might be not the best instrument. τ-ARGUS offers a controlled rounding facility, which is recommended to be used for protecting tables on frequency data.

e) *Single respondent cells* should not be protected by suppression of another single respondent cell. ACS/CONFID and GHQUAR will avoid this situation.

f) Extra care is required, *if units contribute to more than one cell* (along the same axis of a table), as the total of the suppressions along the axis might be predominated by the sum of a single contributors contributions to these suppressed cells. This sum would then be approximately disclosed. CONFID/ACSSuprs and USBCSUP attempt to avoid this situation. τ-ARGUS cannot even represent such tables. For details see [7].

III.4 Input Files and Software Utility

a) *Input files / Output files*

All programs require information on table structure, and information relating to the table cells or micro-data. For brief descriptions regarding input data see [7].

b) *Recoding facilities*

To avoid the release of tables with too many suppressions, the user may (after inspection of first results from the cell suppression procedure) wish to reduce depth of disaggregation. This can be achieved by means of recoding a variable, which is effectively the same as to introduce submarginals, while at the same time suppressing the entire set of internal cells contributing to those submarginals.

Table 6. Input/output files and recording facilities

Software system	Input files / Output files	Recoding facilities
ACS/CONFID	The programs may be run on micro-data, as well as on tabular-data. Output is provided in identical format to the (tabular-data) input, with suppressions suitably flagged. Additionally, all subtables are available in table format.	When using micro-data as input, recoding can be done relatively easily.
τ-ARGUS	Micro-data are required. Output is provided in table format only.	The program was designed for interactive use of cell-suppression, and recoding. The recoding option is comfortable and easy to use. However, local recoding, i.e. to define a variables recoding scheme w.r.t. category of other variables in a multiway-table is not possible.
GHQUAR	The program expects tabular data as input. Output file format is identical to input file format.	Unlike with USBCSUP (see below), generally it is not an option to simply drop certain sets of cells. However, assigning low weights (cf. 3.2) to those cells, which one would like to drop, will actually have a similar effect. A package of utility-routines to make the approach easy, and quick to use is presently in developement.
USBCSUP	The program expects tabular data as input[5)] . The output file is actually identical to the input file, complementary suppressions indicated by flags, and some additional cell related information added.	The program allows for certain cells to be dropped from the table. In hierarchical tables, dropping the entire set of internal cells contributing to certain submarginals would actually be equivalent to recoding. In multi-way tables, the user may even drop only particular subsets of those cells, depending on their hierarchical situation with respect to the other variables, which would in fact be a suitable approach for local recoding. However for larger tables, this recoding approach would require the implementation of additional utility-routines.

c) *Report files*

Report files should contain enough information for the user to understand why certain suppressions / suppression patterns were chosen. We were especially happy with the report files produced by USBCSUP: Depending on the print option, the program presents all subtables treated as single problems during the different stages of the suppression process. A second file provides information on the relations between primary and actually chosen secondary suppressions.

With the assistance of another USBC-program making use of that file, the user can reconstruct the entire suppression pattern used for protection of each single primary suppression. Additionally, statistics of value suppressed and number of cells suppressed are presented.

[5)] Ideally data including information on the two largest contributors to all cells.

IV. SUMMARY

- *ACS/CONFID*: On the class of tables, which the present version of τ-ARGUS cannot be applied to, both these systems perform best regarding information loss. They are most reliable systems with respect to avoidance of disclosure risks, however the application must not exceed moderate size, with dimensions less or equal to three (seven for ACSSuprs). They are probably too slow for applications requiring an iterative process, i.e. larger sets of multiple big tables, or even to single big tables.
- τ-*ARGUS* is a modern, most promising system. It is WINDOWS based, the only system with a user-interface, offering controlled rounding as an alternative technique to cell suppression. The underlying algorithm yields the exact solution of the secondary cell suppression problem, and performed extraordinarily well in experiment (which indeed was a single, relatively small application only). However, the system still requires further developement, and improvement, first of all to make it applicable to tables with hierarchical substructure, to tables with a decomposition structure of the response variable, and to linked tables. To preserve to some extend the systems excellent performance with respect to information loss, while making it capable of solving in acceptable time those large problems, which result from the representation of big real life tables with many submarginals, will be a challenge.
- *GHQUAR* is a very efficient, powerfull system, ideally suited for application to large statistical tables, as well as to multiple tables in an iterative fashion. A utility package is in developement, which will provide facilities to make table-to-table protection for the entire set of multiple tables published on basis of a common data set easily applicable and comfortable.
- *USBCSUP* can be applied to up-to-three dimensional tables with marginals and submarginals. On this class of tables the system performs well regarding information loss, avoidance of disclosure risk, and computing time requirements. A good written manual, a clear source code, which is supplied along with the package, and report files providing enough information to understand a particular suppression process in detail, make it relatively well suited for outside distribution, at least to where FORTRAN knowledge is available.

V. FINAL CONCLUSIONS

17. All five systems solve the secondary cell suppression problem properly. The resulting suppression patterns are fully acceptable, with respect to both: disclosure control, as well as information loss.

18. The most promising system τ-ARGUS still requires further developement to make it applicable to tables with hierarchical substructure and to linked tables. In the meantime, for decision whether or not to procure one of the five software systems to automatize cell suppression procedures, and which to chose, size and structure of the tables to be protected should be considered, while practical aspects may also be taken into account.

APPENDIX 1: AVAILABILITY, PORTABILITY

a) *Availability*

b) *Portability, Hardware Requirements*

The programs store all information on the part of the table actually treated as a single problem in arrays, which may become quite large. Depending on the machine, if memory is limited, this might create a problem, forcing the user to subdivide his application.

Software system	Availability	Portability, Hardware Requirements
ACS	Commercial system, for further information contact *Gordon Sande, Sande & Ass., Inc., 600 Sanderling Court, Secaucus, N.J. 07049, USA*	No detailed information available.
CONFID	The system is available for free at Statistics Canada, on terms to be negotiated on particular request. For further information contact *Dale Robertson (email: Dale.Robertson@statcan.ca), or Jean-Louis Tambay, Statistics Canada, Tunney's Pasture, RHC Building, Ottawa, Ontario, K1A 0T6*	Source language of the program is RATFOR. From this code the preprocessor RATFOR generates FORTRAN-code. This is useful in handling portability issues, making it relatively easy to implement the package on different hardware platforms.
τ-*ARGUS*	Available for free at Statistics Netherlands, in addition to the software an LP-solver is required. Commercial LP-solvers are Xpress (at EUR 900) or Cplex (at EUR 2000). For further information contact *Anco Hundepool (email AHNL@cbs.nl), Department for Statistical Methods, Statistics Netherlands, P.O.Box 4000, 2270 JM Voorburg*	Program was designed for implementation under Windows '95 (or Windows-NT). Installation under Windows 3.11 is possible, together with Win 32s.
GHQUAR	The system is, on particular request, available on terms to be negotiated, at the Statistisches Landesamt für Datenverarbeitung und Statistik Nordrhein-Westfalen. For further information contact *Dietz Repsilber, LDS NRW, Postfach 10 11 05, 40002 Düsseldorf, Germany*	FORTRAN-program, including three (minor) ASSEMBLER-subroutines. The program was developed for implementation on an IBM-mainfraime, and could (slighly modified) be successfully implemented on various mainframes in all German Statistical Offices.
USBCSUP	The system is available for free at the U.S. Bureau of the Census, for further information contact *Laura Zayatz (email:laura.zayatz@ccmail.census.gov), U.S. Bureau of the Census, Washington, DC 20233*	The program has been developed for DEC computers. Running the program on another type of computer may require some small modifications. It has, with little modification in the source code, be successfully implemented on the Siemens mainframe at the Statistisches Bundesamt.

APPENDIX 2: METHODOLOGY APPLIED

19. In this section we assume the reader has some familiarity with the LP formulation of the secondary cell suppression problem. The brief methodological descriptions below are intended to show and emphasize similarities and differences of the various algorithms.

20. The Secondary Cell Suppression problem has been formulated mathematically as Integer Linear Programming (ILP) problem[1]: To all possible complements costs are assigned as to represent the information loss associated with their suppression. The objective function, the sum of the costs of the complements, is minimized subject to a set of constraints, which represent the requested uncertainty about the true values of the primary suppressions, the requirements of maintaining table additivity, and additional constraints such as positivity in the solutions of the linear problem.

ACS/CONFID

21. A Linear Programming approach is used in the CONFID program package of Statistics Canada, and in the commercial package ACSSuprs, which in dead is an improved version of CONFID. The algorithm has been described in [2].

22. With this approach, ideally all cells and their inter-relationships from the entire set of multiple, inter-related tables, disseminated from the same survey, are treated together.
A set of linear equations is generated from the input information on cell inter-relationships.
Successively for each primary suppression the Linear Programming (LP) relaxation of the original ILP (see above) is solved.

23. The costs (for suppression of a cell) are assigned to prefer smaller complements.
The LP-relaxation algorithm tends to prefer solutions with smaller changes in many cells rather than large changes in a small number of cells, which is undesirable. The situation is improved by a second run, the refinement run, which is performed after all primary suppressions have been protected for the first time. Here, only cells suppressed in the first run, may be chosen as complements. Costs are assigned to prefer larger cells, which may make some smaller complements superfluous.

τ-ARGUS

24. The Integer Linear Programming approach described in [4] has been implemented in the second version of the program τ-ARGUS. The approach yields the exact solution of the Integer Linear Programming problem. LP-relaxations of subproblems are solved in iterative fashion using effective separation algorithms. For fractional solutions of the LP-relaxation the algorithm branches into new subproblems.

GHQUAR

25. The method has been described in [5]. The approach subdivides n-dimensional tables with marginals and submarginals into a set of n-dimensional subtables without submarginals. These subtables are protected successively within an iterative procedure, starting from the highest level. Successively for each primary suppression in the current subtable, all possible hypercubes with this cell as one of the corner points are constructed.

26. Though avoiding the solution of time consuming LP-problems, for each hypercube a lower bound is calculated for the amount of protection (namely the sum of potential decrease, and potential increase of the cell value) that suppression of all its corner points would give to the primary suppression.

27. If this bound is sufficiently large, the hypercube becomes a feasible solution.
Among all feasible solutions those hypercubes which require a minimum number of additional suppressions are considered. The one which leads to a minimum loss of information associated with the suppression of its corner points is selected. After all subtables have been protected once, the procedure is repeated in an iterative fashion. Complements of subtotals are carried over from subtable to subtable and protected iteratively.

USBCSUP

28. A network-flow approach as suggested for example in [3] is used within USBCSUP.
The table is subdivided into two dimensional subtables with tree substructures only in the columns. These 2D-subtables are converted into networks. The arcs of the network correspond to the cells of the subtable. For each primary suppression in the subtable a minimal flow closed path is selected by solving the LP relaxation of the ILP problem. A refinement run is done for each target suppression to be protected, which may make some smaller complements superfluous.

References

[1] J. Geurts, Heuristics for Cell Suppression in Tables, working paper, Netherlands Central Bureau of Statistics, 1992

[2] D. Robertson, Automated Disclosure Control at Statistics Canada, paper presented at the Second International Seminar on Statistical confidentiality, Luxemburg, 1994

[3] L. Cox, Solving Confidentiality Problems in Tabulations Using Network Optimization: A Network Model for Cell Suppression in the U.S. Economic Censuses, Proceedings of the International Seminar on Statistical confidentiality, Dublin, 1992

[4] M. Fischetti and J.J. Salazar, Modelling and Solving the Cell Suppression Problem for Linearly-Constrained Tabular Data, In J. Domingo-Ferrer, ed., Statistical Data Protection '98, Conference Proceedings 25-27 March 1998, Lisbon, Portugal

[5] D. Repsilber, Preservation of Confidentiality in Aggregated data, paper presented at the Second International Seminar on Statistical Confidentiality, Luxemburg, 1994

[6] G. Sande, Sande & Associates, Inc.,various internal papers

[7] S. Gießing, A Comparison of Automated Secondary Cell Suppression Systems, Federal Statistical Office of Germany, working paper, 1998

[8] D. Repsilber, Wahrung der Geheimhaltung in aggregierten Daten
- Quaderverfahren mit Intervallschutz für vollständige Tabellen - in Forum der Bundesstatistik, Methoden zur Sicherung der Statistischen Geheimhaltung, to appear , 1998

[9] K. McLeod / J. George / A. Ray / R. Butler, Investigating key Qualities of an Automated Cell Suppression System, Proceedings of the SDP-Conference, to appear, 1998

[10] L. Cox, Linear Sensitivity Measures in Statistical Disclosure Control, Journal of Planning and Inference, 5, 153 - 164, 1981

[11] D. Robertson, Cell Suppression at Statistics Canada, Proceedings Annual Research Conference, U.S. Bureau of the Census, 1993

Acknowledgements

The author is grateful to Laura Zayatz and Bob Jewett, for supplying the USBCSUP software, and to Dale Robertson for running CONFID on tables from the German Census of Manual Trades.