How to Make the τ-ARGUS Modular Method Applicable to Linked Tables

Peter-Paul de Wolf[1] and Sarah Giessing[2]

[1] Statistics Netherlands,
Department of Methodology and Quality,
P.O. Box 24500
2490 HA Den Haag, The Netherlands
PWOF@cbs.nl
[2] Federal Statistical Office of Germany,
65180 Wiesbaden, Germany
Sarah.Giessing@destatis.de

Abstract. The software package τ-ARGUS offers a very efficient algorithm for secondary cell suppression known as either HiTaS or the Modular approach. The method is well suited for the protection of up to 3-dimensional hierarchical tables. In practice, statistical agencies release multiple tabulations based on the same dataset. Usually these tables are linked through certain linear constraints. In such a case cell suppressions must obviously be coordinated between tables. In this paper we investigate into the possibilities for an extension of the modular approach to deal with linked tables.

1 Introduction

Some cells of the tabulations released by official statistics contain information that chiefly relates to single, or very few respondents which may often be easily identifiable. Therefore, traditionally, statistical agencies suppress part of the data, hiding some table cells from publication. Efficient algorithms for cell suppression are offered e.g., by the software package τ-ARGUS [6].

When tables are linked through simple linear constraints, cell suppressions must obviously be coordinated between these tables. A frequently occurring instance of a set of linked tables, consists of tables that share some of their marginal cells. E.g., tables specified in Eurostats SBS-regulation: a table on turnover broken down by 4-digit NACE, a second table on turnover broken down by 3-digit NACE and size class and a third table on turnover broken down by 2-digit NACE and geographic location (NUTS). These tables obviously have some marginal cells in common.

The intention of this paper is to present a collection of several alternative approaches for this coordination problem, and to give an idea of the issues that have to be considered for a decision which of the approaches (if any) should eventually get implemented in τ-ARGUS within the framework of a current joint European cooperation project.

Considering correctly the links between tables in the cell suppression process leads to a substantial increase in problem complexity. This tends to lead in turn to substantial increase in the amount of information that will be suppressed. One way of avoiding at least part of this increase may be to improve certain mechanisms in the current heuristics of the Modular method which cause overprotection in some situations. Section 4.1 proposes an idea for how to improve current heuristics.

2 Methodological Background

Statistical offices collect information on several properties that might be used for grouping respondents, like e.g. information about respondent economic activity (NACE) and geographic location. While with modern technologies it is no problem anymore to generate any kind of tables, or, by means of data-warehousing systems, to allow users to construct their own tabulations, solving the corresponding disclosure control problems consistently by means of secondary cell suppression can hardly be achieved in full generality because of the problem of coordinating suppressions across linked tables. [3] has described a special class of linked tables and presented an idea for an extension of the current methodologies to deal with sets of linked tables belonging to this class. This class of linked tables includes the linked tables as specified in the SBS-regulation of Eurostat.

The next subsection will first introduce some definitions and denotations which we will use throughout the paper with respect to hierarchical and linked tables structures. Subsection 2.2 briefly describes the original modular method (a.k.a. HiTaS). Finally, in subsection 2.3 we will discuss four possible approaches to deal with sets of linked tables.

2.1 Definitions and Denotations

In the terminology of tabular data statistical disclosure control, we think of an *m*-dimensional table as a tabulation of a certain continuous *response* variable by a cross combination of *m* categorical *spanning* variables.

[3] has introduced some denotation on hierarchical structures between the categories of spanning variables taken from graph theories. We follow this denotation in this paper and consider a hierarchy to be a rooted, directed tree, with the categories being the vertices of the tree. Additionally we define:

- A *relation* is a hierarchy consisting of only one father vertex and the corresponding child vertices.
- A table given as a cross combination of relations ($\mathcal{R}_1 \times \ldots \times \mathcal{R}_m$) is called a *simple table*. Note that this kind of table is often referred to as 'non-hierarchical' or 'unstructured'.
- If $\mathcal{G}$ is the covering hierarchy of a set of relations $\{\mathcal{R}_1,\ldots,\mathcal{R}_k\}$ then we say that $\{\mathcal{R}_1,\ldots,\mathcal{R}_k\}$ is a *simple breakdown* of $\mathcal{G}$.
- The *level* of a relation $\mathcal{R}_j$ in a simple breakdown of a hierarchy $\mathcal{G}$ is the level of the root of $\mathcal{R}_j$ in $\mathcal{G}$.

- Without loss of generality, we define the level of $\mathcal{R}_1$ to be 0. Note that for any $j > 1$ the root category of $\mathcal{R}_j$ is also a category of $\mathcal{R}_l$ for some $l \neq j$. We then say that $\mathcal{R}_j$ and $\mathcal{R}_l$ are *linked*.

Consider an m-dimensional table T given as ($\mathcal{G}_1 \times \ldots \times \mathcal{G}_m$). Let $\{\mathcal{R}_1^i,...,\mathcal{R}_{k_i}^i\}$ denote the simple breakdown of $\mathcal{G}_i$. The *breakdown of table T into simple (sub)tables* is then given as the set $S(T)$ of up to m-dimensional simple tables $T_{j_1 j_2 \ldots j_m}$. Each of the simple tables $T_{j_1 j_2 \ldots j_m}$ is given as cross combination of relations $(\mathcal{R}_{j_1}^1 \times \mathcal{R}_{j_2}^2 \times ... \times \mathcal{R}_{j_m}^m)$ with $1 \leq j_i \leq k_i$. Because some of the $\mathcal{R}_{j_i}^i$ are linked, some of the tables $T_{j_1 j_2 \ldots j_m}$ in the set $S(T)$ are linked, i.e., they share identical cells.

2.2 The Original Modular Approach for Dealing with Hierarchical Tables

The disclosure risk connected to each individual cell of a table is assessed by applying certain sensitivity rules. If a cell value reveals too much information on individual respondent data, it is considered *sensitive*, and must not be published. We consider this to be the case, if the cell value could be used, in particular by any of the respondents, to derive an estimate for a respondent's value that is closer to the reported value of that unit than a pre-specified percentage p (this sensitivity rule is called the p% rule).

Cell suppression comprises of two steps. In a first step, sensitive cells will be suppressed (primary suppressions). In a second step, other cells (so called secondary suppressions) are selected that will also be excluded from publication in order to prevent the possibility that users of the published table would be able to recalculate primary suppressions. Naturally, this causes an additional loss of information.

By solving a set of equations implied by the additive structure of a statistical table, and some additional constraints on cell values (such as non-negativity) it is possible to obtain a *feasibility interval*, i.e., upper and lower bounds for the suppressed entries of a table, c.f. [4], for instance. A set of suppressions (the '*suppression pattern*') is called '*valid*', if the resulting bounds for the feasibility interval of any sensitive cell cannot be used to deduce bounds on an individual respondent's contribution that are too close according to the criterion employed to assess cell sensitivity. This requires that the bounds of the feasibility interval for any sensitive cell are at a 'safe' distance from its true value. Safe distance means that the distance exceeds the so called *protection level,* c.f. [7, 4.2.2].

The problem of finding an optimum set of suppressions known as the 'secondary cell suppression problem' is to find a feasible set of secondary suppressions with a minimum loss of information connected to it. The 'classical' formulation of the secondary cell suppression problem leads to a combinatorial optimization problem, which is computationally extremely hard to solve. For practical applications, the formulation of the problem must be relaxed to some degree.

The modular approach for hierarchical table cell suppression (also called HiTaS, see [2] for a detailed description) subdivides a hierarchical table T into the corresponding set $S(T)$ of simple, 'unstructured' linked (sub-)tables. The cell

suppression problem is solved for each subtable separately. Within each subtable, methods based on Fischetti/Salazar Linear Optimization tools [4] are used to select secondary suppressions. For the co-ordination of secondary suppressions between linked subtables a backtracking procedure is used: the modular approach deals with the tables in $S(T)$ in a special, ordered way. During processing it notes any secondary suppression belonging also to one of the other tables. It will then suppress it in this table as well, and eventually repeat the cell suppression procedure for this table.

It must however be stressed, that a backtracking procedure is not global according to the denotation in [1]. See [1] for discussion of problems related to non-global methods for secondary cell suppression.

2.3 Extension of the Modular Approach for Dealing with Linked Tables

[3] presents an idea to extend the current methodologies to deal with a set of linked tables $\{T_1,\ldots,T_N\}$ that satisfy certain criteria. For instance, it is assumed that each table has a hierarchical structure that may differ from the hierarchical structures of the other tables. However, it is also assumed that tables that use the same spanning variables only have hierarchies that can be covered by a single hierarchy for that spanning variable. See [3] for the definition of a covering hierarchy. In essence it means that the covering hierarchy is such that all related hierarchies can be viewed as sub-hierarchies.

In the context of pre-planned table production processes which are typically in place in statistical agencies for the production of certain sets of pre-specified standard tabulations, it is normally no problem to satisfy these conditions. Literally speaking, the assumption is that tables in a set of linked tables may present the data in a breakdown by the same spanning variable at various amount of detail. But only under the condition that, if in one of the tables some categories of a spanning variable are grouped into a certain intermediate sum category, during SDC processing this intermediate sum category is considered in any other table presenting the data in a breakdown of the same spanning variable and at that much detail.

The idea of [3] is then as follows. For N tables $\{T_1,\ldots,T_N\}$ that need to be protected simultaneously, suppose that the specified tables contain M different spanning variables. Since the hierarchies are supposed to be coverable, an M-dimensional table exists having all the specified tables as subtables. The spanning variables will be numbered 1 up to M.

Each spanning variable can have several hierarchies in the specified tables. Denote those hierarchies for spanning variable i by $\mathcal{H}_1^i,\ldots,\mathcal{H}_{I_i}^i$ where I_i is the number of different hierarchies.

Define the M-dimensional table by the table with spanning variables according to hierarchies $\mathcal{G}_1,\ldots,\mathcal{G}_M$ such that, for each $i = 1,\ldots, M$ hierarchy $\mathcal{G}_i$ covers the set of hierarchies $\{\mathcal{H}_j^i\}$ with $j = 1,\ldots, I_i$. This M-dimensional table will be called the cover table.

We will now describe several approaches to deal with this set of linked tables.

Complete Modular Approach
A straightforward approach would be to protect the complete cover table. HiTaS deals with all possible simple ('non-hierarchical') subtables of a hierarchical table in a specially ordered way. This would take care of all links between the tables in the set $\{T_1,\ldots,T_N\}$, since by definition these tables are subtables of this cover table. This would result in a set of N protected tables $\{T_1^P,\ldots,T_N^P\}$. However, this approach considers a table structure which is much more complex than that of the tables which actually get published. We expect that this will tend to lead to a substantial increase in information loss compared to the other methods. Moreover, primary suppressions at low levels of a hierarchy often lead to secondary suppressions at the higher levels. Hence, unsafe cells in detailed tables that will not be published, might lead to secondary suppressions in less detailed tables that will be published. These secondary suppressions may be considered to be superfluous. This is probably not acceptable to users, given that the actual disclosure risk caused by ignoring subtables that are not foreseen for publication during disclosure control is likely to be rather low.

Adapted Modular Approach
The modular approach of HiTaS can easily be adapted. The idea is now basically to use the modular approach on the cover table T_C, but only consider those subtables that are also subtables of at least one of the specified tables $T_1,\ldots,T_N$ and disregard the other subtables. In the following we denote this subset of $S(T_C)$ (the breakdown of cover table T_C into subtables) as $S^*(T_C)$. This approach was suggested in [3] as well.

In τ-ARGUS the original modular approach is limited to hierarchical tables with up to three dimensions. This is mainly due to the fact that the Fischetti/Salazar Linear Optimization tools get too slow when dealing with higher dimensional tables. For the Adapted Modular Approach this restriction can be weakened. In theory there is no restriction on the number of dimensions of the cover table T_C, as long as each (sub)table that needs to be protected is at most three dimensional.

Linked Subtables Modular Approach
This somewhat more complex approach deals with sets of linked, simple subtables at a time. For each table T_i construct the set of linked subtables $S(T_i)$. Then consider the ordering used in HiTaS to order each set $S(T_i)$. Then deal with subtables from $S(T_i)$, …, $S(T_N)$ that are on the same order-level as linked tables using Fischetti/Salazar Linear Optimization tools. Such a set of linked subtables on the same order level is constructed in the following way: Let U and V be two simple subtables in $S^*(T_C)$. Assume U is a ν-dimensional (simple) table, where the first n spanning relations of U are not at level 0 of the corresponding covering hierarchies, i.e., U can be represented as $U := (\mathcal{R}_{i_1}^{1} \times \ldots \times \mathcal{R}_{i_n}^{n} \times \mathcal{R}_{1}^{n+1} \times \ldots \times \mathcal{R}_{1}^{v})$. Then the subtable V belongs to the same set of linked subtables, if it is based on the same first n spanning relations as U, i.e., if it can be stated as $V := (\mathcal{R}_{i_1}^{1} \times \ldots \times \mathcal{R}_{i_n}^{n} \times \mathcal{R}_{1}^{n+j} \times \ldots \times \mathcal{R}_{1}^{n+l})$ for some $M \geq l \geq j \geq 0$ where M is the dimension of the cover table T_C.

If all spanning relations of U are at level 0, i.e., $n = 0$, then the condition is that U and V share at least one level-0 spanning relation which can be expressed formally by requiring $j = 1$.

Traditional Approach

Although HiTaS cannot (yet) deal with linked tables, statistical agencies using HiTaS for secondary suppression of single tables must somehow solve the co-ordination problem. One possible approach is discussed in [7, 4.3.3]. This 'traditional' method is based on the idea of a backtracking procedure on the table level instead of on the sub-table level.

In case of two linked tables T_1 and T_2, the approach would be as follows:

1. Protect table T_1 on its own;
2. Each cell in T_2 that is also present in T_1 will get the status (i.e., suppressed or not-suppressed) of the cell in the protected table T_1;
3. Table T_2, with the additional suppressions carried over in step 2, is protected on its own.
4. Each cell in T_1 that is also present in T_2 will get the status of the cell in the protected table T_2;
5. Repeat step 1 – 4 until no changes occur in protecting table T_1 nor in protecting T_2.

Graphically this would look like Figure 1.

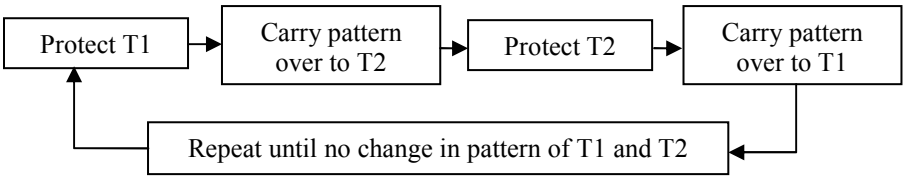

Fig. 1. Graphical representation of iteratively protecting two linked tables.

Adding a third table to the set of linked tables, i.e., considering $\{T_1, T_2, T_3\}$, adds some complexity to this procedure. In that case several schemes can be thought of. E.g.,

a. Protect T_1, carry pattern over to T_2, protect T_2, carry pattern over to T_3, protect T_3, carry pattern over to T_1, repeat until no changes are added.
b. Protect T_1, carry pattern over to T_2, protect T_2, carry pattern over to T_1, protect T_1, repeat until no changes in $\{T_1, T_2\}$, carry patterns over to T_3, protect T_3, carry pattern over to T_1 and T_2, start with T_1 again. Repeat until no changes in T_1, T_2 and T_3.

The choice to be made may depend on the structure of the links between the tables T_1, T_2 and T_3. Obviously, the more linked tables need to be considered simultaneously, the more schemes can be constructed.

3 Illustrative Examples

In this section we will demonstrate the different approaches explained in the previous section using some instructive examples.

Example 1
We first consider a very simple instance of two linked tables. The specification of the two tables involves three spanning variables $\mathcal{F}$, $\mathcal{G}$ and $\mathcal{H}$, where $\mathcal{F}$ and $\mathcal{H}$ each consist of one relation only. The first table is given as $\mathcal{G}_2 \times \mathcal{F}$, the second table as $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$. The simple breakdown of $\mathcal{G}_1$ consists of three relations, $\mathcal{R}_1^{\mathcal{G}}$, $\mathcal{R}_2^{\mathcal{G}}$ and $\mathcal{R}_3^{\mathcal{G}}$, where $\mathcal{R}_2^{\mathcal{G}}$ and $\mathcal{R}_3^{\mathcal{G}}$ are at level 1. The simple breakdown of $\mathcal{G}_2$ is given by ($\mathcal{R}_1^{\mathcal{G}}$, $\mathcal{R}_2^{\mathcal{G}}$, $\mathcal{R}_3^{\mathcal{G}}$, $\mathcal{R}_4^{\mathcal{G}}$, …, $\mathcal{R}_{20}^{\mathcal{G}}$) where $\mathcal{R}_4^{\mathcal{G}}$ to $\mathcal{R}_{10}^{\mathcal{G}}$ are at level 1 and $\mathcal{R}_{11}^{\mathcal{G}}$ to $\mathcal{R}_{20}^{\mathcal{G}}$ are at level 2. So $\mathcal{G}_1$ is a pure subhierarchy of $\mathcal{G}_2$, and therefore $\mathcal{G}_2$ is the covering hierarchy for variable $\mathcal{G}$.

The set $S^*(T_C)$ of subtables of the cover table T_C that are also subtables of T_1 or T_2, is then given by $\{\mathcal{R}_i^{\mathcal{G}} \times \mathcal{F} \times \mathcal{H}$, with $i = 1, \ldots, 4\} \cup \{\mathcal{R}_i^{\mathcal{G}} \times \mathcal{F}$, with $i = 5, \ldots, 20\}$.

According to the Adapted Modular Approach, we deal with these subtables successively within the usual backtracking strategy of HiTaS. The Linked Subtables Modular Approach is in this simple instance identical to the Adapted Modular Approach.

For the traditional approach we start with the first table $\mathcal{G}_2 \times \mathcal{F}$, use HiTaS for secondary suppression, carry the secondary suppressions from the area where both tables overlap, i.e., $\{\mathcal{R}_i^{\mathcal{G}} \times \mathcal{F}$, with $i = 1, \ldots, 4\}$, over to the second table $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$, do secondary suppression with HiTaS, and carry new secondary suppressions in $\{\mathcal{R}_i^{\mathcal{G}} \times \mathcal{F}$, with $i = 1, \ldots, 4\}$ over to the first table. In the instance the first table could then be processed successfully without selecting any new secondary suppressions in $\{\mathcal{R}_i^{\mathcal{G}} \times \mathcal{F}$, with $i = 1, \ldots, 4\}$ and thus the process finished successfully.

The results are summarized in Table 1. For a more detailed presentation of the results see Table 2 in the appendix.

The table $\mathcal{G}_2 \times \mathcal{F}$ contained 72 primary unsafe cells, 26 empty cells and 1343 cells in total. Table $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$ consisted of 4896 cells of which 657 cells were primary unsafe and 1055 were empty. The costs for suppressing a cell was defined to be $a_i \wedge 0.4$ with a_i the cell value.

Table 1. Results of running three approaches to protect a set of two linked tables

Approach*	Number of secondary suppressions		Sum of costs of secondary suppressions	
	$\mathcal{G}_2 \times \mathcal{F}$	$\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$	$\mathcal{G}_2 \times \mathcal{F}$	$\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$
ModFull	96	709	8420	33330
ModAd	73	677	6528	31234
Trad	75	788	6440	34452

* ModFull = Complete Modular, ModAd = Adjusted Modular, Trad = Traditional

In this (rather small) instance, with regard to the number of suppressions, the Adapted Modular (ModAd) approach outperforms the other two, i.e., the Complete Modular (ModFull) and the Traditional (Trad) approach. Table 2 (Appendix) presents the same results at some more detail by hierarchical level of the spanning variables. While the results of the traditional method on table $\mathcal{G}_2 \times \mathcal{F}$ are quite reasonable, the method performs especially bad at the second level of variable $\mathcal{F}$ in the $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$ table. This is perhaps a consequence of running $\mathcal{G}_2 \times \mathcal{F}$ first. Note that in this instance the disclosure risk for the suppression pattern provided by the traditional method is likely to be higher compared to patterns resulting from the other two approaches, because we decided to protect secondary suppressions carried over from the other table against exact disclosure only, assigning constant protection levels of 1 to those cells. This is probably also the reason why the sum of costs of secondary suppressions in $\mathcal{G}_2 \times \mathcal{F}$ is smaller for the traditional approach, compared to the adapted modular, even though the number of secondary suppressions is larger (73 vs. 75 cells). For more discussion on protection levels for secondary suppressions see 4.1.

Example 2

In this example, we add a third table to the two tables of example 1. This third table is given by $\mathcal{G}_2 \times \mathcal{H}$. For the cover table T_C of example 2 it holds:

$S^*(T_C) = \{ \mathcal{R}_i^{\mathcal{G}} \times \mathcal{F} \times \mathcal{H}$, with $i = 1, \ldots, 4\} \cup \{ \mathcal{R}_i^{\mathcal{G}} \times \mathcal{F}$, with $i = 5, \ldots, 20\} \cup \{ \mathcal{R}_i^{\mathcal{G}} \times \mathcal{H}$, with $i = 5, \ldots, 20\}$.

The extended set of subtables involving some pairs (U, V) of linked subtables for the Linked Tables Modular Approach is then given by $S^{**}(T_C) = \{ \mathcal{R}_i^{\mathcal{G}} \times \mathcal{F} \times \mathcal{H}$, with $i = 1, \ldots, 4\} \cup \{(\mathcal{R}_i^{\mathcal{G}} \times \mathcal{F}, \mathcal{R}_i^{\mathcal{G}} \times \mathcal{H})$, with $i = 5, \ldots, 20\}$.

For selecting secondary suppressions assigned earlier in the process in a table T_j to be carried over to a table T_i $(i \neq j)$ the areas $T_i \bigcap T_j$ have to be considered. Note that in each step (but only after processing the second table for the first time), we have to consider two of those areas, e.g., $T_1 \bigcap T_3$ and $T_2 \bigcap T_3$ for importing secondary suppressions to the third table. In our instance the overlap areas are $T_1 \bigcap T_2 = \{ \mathcal{R}_i^{\mathcal{G}} \times \mathcal{F} \times \mathcal{H}$, with $i = 1, \ldots, 4\}$, $T_1 \bigcap T_3 = \{ \mathcal{R}_i^{\mathcal{G}}$ with $i = 1, \ldots, 20\}$ and $T_2 \bigcap T_3 = \{ \mathcal{R}_i^{\mathcal{G}} \times \mathcal{H}$, with $i = 1, \ldots, 4\}$.

Tables 3 and 4 in the appendix show the results of processing these 3 tables by the adapted modular and the traditional approach. The performance of the methods is similar as observed for instance 1: the adapted modular method gave superior results, especially for the 3-dimensional table.

Example 3

In this example we discuss the special case, where a table in a set of linked tables presents results for a subpopulation only, while other tables in the set present results on the full population. The instance this time consists of the two tables of example 1, $\mathcal{G}_1 \times \mathcal{F}$ and $\mathcal{G}_2 \times \mathcal{F} \times \mathcal{H}$ with a third table added, which is this time given by $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}|_5$. This third table presents data on the subpopulation falling into category 5 of hierarchy $\mathcal{H}$. We denote this special 'subhierarchy' of $\mathcal{H}$ consisting of only one category as $\mathcal{H}|_5$.

Hierarchy $\mathcal{H}$ must then be extended, since we now consider two relations. The first one, $\mathcal{H}_1$, defines the partition of the full population into category 5 and a 'rest'-category consisting of all categories of $\mathcal{H}$ except category 5. The second one, $\mathcal{H}_2$, describes the partition of that 'rest' into the other categories. $\mathcal{H}$* denotes then the covering hierarchy of $\mathcal{H}_1$ and $\mathcal{H}_2$. We can now re-specify the second table as $\mathcal{G}_2 \times \mathcal{F} \times \mathcal{H}$* and the second and third table can be joined into one, e.g., $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}_1$. In this way we avoid certain disclosure-by-differencing problems. For instance, if for a given category g* of $\mathcal{G}$ only one respondent falls into the 'rest' category of $\mathcal{H}_1$, but the two cells specified by the two other categories of $\mathcal{H}_1$ (e.g., category 5 and the root-category) happen to be safe and remain unsuppressed. Such a problem will only be detected by a disclosure control process that explicitly considers the 'rest'.

On the other hand, in practice we sometimes deal with much more than just three linked tables. Taking into account correctly the relations between subpopulations considered for publication and subpopulations not considered for population usually adds to the complexity of the problem. In order to keep the effort for disclosure control processing within reasonable limits, in practice such subpopulation relations are often ignored. This can be justified, if the resulting disclosure risks are rather low, which is typically the case if the subpopulation of interest is comparatively small.

4 A Special Application of Partial Suppression Technology

In section 2.2 we have mentioned that on the level of individual simple sub-tables HiTaS uses methods based on Fischetti/Salazar Linear Optimization tools [4] to select secondary suppressions. In [5] the same authors propose a relaxed technique. The *complete cell suppression* method of [4] selects among all feasible suppression patterns the one with minimum information loss. This is modeled by associating a weight w_i with each cell i of the table and by requiring the minimization of the overall weight of the suppressed cells, e.g., it minimizes $\sum_{\{sup\}} w_i$ where $\{sup\}$ is the set of suppressed cells. The idea of the *partial cell suppression* methodology of [5] is, on the other hand, to compute intervals around the true cell values a_i , $[a_i - z_i^-, a_i + z_i^+]$, say. A set of intervals is considered as feasible, if the feasibility intervals for sensitive cells that could be computed taking into account the linear relations of the table and the intervals supplied by the partial suppression method cover certain pre-defined protection intervals. Based on the assumption that in a publication the true cell values would be replaced by these intervals, the loss of information associated to such a replacement is modeled as the size of the interval, i.e., $z_i^- + z_i^+$, or, in a more flexible way, as weighted linear combination of the deviation between interval bounds and true cell value $w_i^- z_i^- + w_i^+ z_i^+$. Seeking to minimize the overall information loss then means to minimize $\sum (w_i^- z_i^- + w_i^+ z_i^+)$.

Although the partial cell suppression problem is computationally much easier to solve compared to the complete cell suppression problem, and although a prototypical implementation for the partial suppression approach exists, in practice it has not yet been tested so far. Statistical agencies tend to be rather reluctant to replace traditional cell suppression by an interval publication strategy. Of course the strategy could be to

suppress all cells where $z_i^- + z_i^+$ is non-zero. However, the set of these cells tends to be much larger then the set of cells suppressed as a result of the complete suppression approach.

In this paper we propose now a strategy to use partial cell suppression as complementary technique for complete cell suppression within the backtracking procedure of the modular approach. Obviously, we could also use this idea when we are carrying over suppression patterns between linked tables.

4.1 Using Partial Suppression to Compute Protection Levels

In 2.2 it was mentioned that a suppression pattern for a subtable is considered valid, only if the bounds of the feasibility interval for any sensitive cell are at a 'safe' distance from its true value, exceeding the protection level of that cell. Suitable protection levels are computed by τ-ARGUS according to [7, 4.2.2, table 4.2] and depend on the distribution of the individual contributions to a sensitive cell.

HiTaS deals with each subtable separately, carrying over secondary suppressions from overlapping subtables. A suppression pattern for a subtable T_a with secondary suppressions 'imported' from other subtables should be considered valid only, if the feasibility interval for a secondary suppression imported from, say, a subtable T_b does not jeopardize the protection provided to the sensitive cells of subtable T_b. Assume an intruder first computes the feasibility intervals for all suppressed cells of the subtable T_a. Later in his analysis, the intruder is assumed to consider these feasibility intervals as *a priori* bounds when computing feasibility intervals for suppressed cells of subtable T_b. Even with this additional *a priori* information, the resulting feasibility interval for the sensitive cells of subtable T_b should still be sufficiently wide. In the current implementation of HiTaS this issue is addressed by assigning protection levels to secondary suppressions computed by means of a simple heuristic. The following considerations aim at the development of a theoretically sound methodology to replace this heuristic.

Partial suppression provides us with a set of (smallest) intervals which can be published safely. This means that feasibility intervals for sensitive cells computed considering the partial suppression intervals as *a priori* bounds are sufficiently wide. It will therefore be enough to require for any suppression s in a subtable T_a which is an imported suppression in another subtable T_b that the feasibility interval for s computed on the basis of a suppression pattern for T_b covers the partial suppression interval of s in subtable T_a.

We therefore propose the following strategy:

(1) Compute a suppression pattern for subtable T_a using complete suppression.

(2) Compute a partial suppression pattern for subtable T_a where only cells suppressed in (1) are eligible for (partial) suppression.

(3) Assign the distances between the bounds of intervals given by the partial suppression pattern and the cell value of any suppressed cell s of T_a as protection level to s when protecting any other subtable T_b containing cell s.

Note that this strategy principally may require that protection levels of primary suppressions may have to be changed during processing.

The same strategy could also be used when dealing with the linked table settings of the current paper. E.g., in the traditional approach, the suppression pattern of the first table is carried over to the second (linked) table. If we then want to protect the second table, we will have to treat the complete suppression pattern as primary suppressions. Hence, we will need to specify safety ranges to each suppressed cell. We could use the above proposed strategy to calculate those safety ranges.

5. Summary and Final Conclusions

This paper has presented a few ideas for a backtracking algorithm that might be implemented in order to extend the τ-ARGUS Modular method, making it able to deal with sets of linked tables. We have used small illustrative examples for a first comparison of the algorithm properties. In this first comparison a method outlined in [3], here referred to as 'Adapted Modular Approach', gave promising results.

Another approach, denoted here as 'Linked Subtables Modular Approach' has certain theoretical advantages but is on the other hand more complex. Which of the two works better in practice is a question that we will be able to answer only after some testing on much larger datasets as we have used in this paper. For a decision on which algorithm to implement in a future version of τ-ARGUS the results of such a comparison should be considered.

The challenge of making the τ-ARGUS Modular method applicable to linked tables is, however, not only a matter of finding a good way for subtable construction and ordering sequences for the backtracking. To handle sets of linked tables means also to handle much larger datasets than just single tables. It means that more complex data structures are considered for disclosure control, and this tends to increase the information loss. In order to address these issues, in section 4 we have discussed partial cell suppression methodology. We have drafted a method to determine protection levels for secondary suppressions in a theoretically sound way using partial suppression methodology. This may eventually help to improve the performance of the Modular method regarding information loss.

Acknowledgements

This research was partially supported by the European Commission under grant agreement No 25200.2005.003-2007.670: 'ESSnet on statistical disclosure control'

References

1. 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
2. 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)
3. De Wolf, P.P. (2007), 'Cell suppression in a special class of linked tables', paper presented at the Joint ECE/Eurostat Worksession on Statistical Confidentiality in Manchester, December 2007, available at http://epp.eurostat.ec.europa.eu/portal/page?_pageid=3154,70730193,3154_70730647&_dad=portal&_schema=PORTAL.
4. Fischetti, M, Salazar Gonzales, J.J. (2000), 'Models and Algorithms for Optimizing Cell Suppression in Tabular Data with Linear Constraints', in *Journal of the American Statistical Association*, Vol. 95, pp 916
5. Fischetti, M, Salazar Gonzales, J.J. (2005), 'A Unified Mathematical Programming Framework for different Statistical Disclosure Limitation Mehtods', in *Operations research 53/5* pp. 819-829
6. Hundepool, A., van de Wetering, A., Ramaswamy, R., de Wolf, P.P., Giessing, S., Fischetti, M., Salazar, J.J., Castro, J., Lowthian, P. (2006), τ-ARGUS users's manual, version 3.1.
7. Hundepool, Anco, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Rainer Lenz, Jane Longhurst, Eric Schulte Nordholt, Giovanni Seri, Peter-Paul De Wolf (2006), *CENEX handbook on Statistical Disclosure Control,* CENEX-SDC project, http://neon.vb.cbs.nl/cenex/CENEX-SDC_Handbook.pdf

Appendix

Table 2. Results for Instance 1: Number of suppressed cells for test-tables $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$ and $\mathcal{G}_2 \times \mathcal{F}$ by hierarchy levels[1]

Test-table	Level spanning variable		ModAd	ModFull	Trad
	$\mathcal{G}$	$\mathcal{F}$	# Secondary suppressions		
$\mathcal{G}_2 \times \mathcal{F}$	1	1	0	0	0
	1	2	0	0	0
	2	1	0	0	0
	2	2	10	12	8
	3	1	0	0	0
	3	2	15	26	17
	4	1	0	0	0
	4	2	48	58	50
	All		73	96	75
$\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$	1	1	0	0	0
	1	2	3	3	3
	2	1	7	7	7
	2	2	270	292	357
	3	1	17	23	17
	3	2	355	346	379
	All		652	671	763

[1] For $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$ statistics computed only for cells which are not in $\mathcal{G}_2 \times \mathcal{F}$

Table 3. Results for the set of the three linked tables of Instance 2

Approach	Number of secondary suppressions			Sum of costs of secondary suppressions		
	$\mathcal{G}_2 \times \mathcal{F}$	$\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$	$\mathcal{G}_2 \times \mathcal{H}$	$\mathcal{G}_2 \times \mathcal{F}$	$\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$	$\mathcal{G}_2 \times \mathcal{H}$
ModFull	96	709	101	8420	33330	8413
ModAd	73	710	77	6528	32178	6432
Trad	76	794	79	6484	34077	6158

Table 4. Results for Instance 2: Number of suppressed cells for test-tables $\mathcal{G}_2 \times \mathcal{F}$, $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$ and $\mathcal{G}_2 \times \mathcal{H}$ by hierarchy levels[2]

Test-table	**Level spanning variable**			**ModAd**	**ModFull**	**Trad**
	$\mathcal{G}$	$\mathcal{F}$	$\mathcal{H}$	# Secondary Suppressions		
$\mathcal{G}_2 \times \mathcal{F}$	1	1	1	0	0	0
	1	2	1	0	0	0
	2	1	1	0	0	0
	2	2	1	10	12	9
	3	1	1	0	0	0
	3	2	1	15	26	17
	4	1	1	0	0	0
	4	2	1	48	58	50
	All			73	96	76
$\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$	1	1	2	0	0	0
	1	2	2	3	3	3
	2	1	2	7	7	8
	2	2	2	270	292	360
	3	1	2	16	23	18
	3	2	2	389	346	379
	All			685	671	768
$\mathcal{G}_2 \times \mathcal{H}$	4	1	2	54	71	53

[2] For $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$, statistics based only on cells which are not in $\mathcal{G}_2 \times \mathcal{F}$. For $\mathcal{G}_2 \times \mathcal{H}$, statistics based on cells neither in $\mathcal{G}_2 \times \mathcal{F}$, nor in $\mathcal{G}_1 \times \mathcal{F} \times \mathcal{H}$.