Working Paper
ENGLISH ONLY

UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE (UNECE)

CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION

STATISTICAL OFFICE OF THE EUROPEAN UNION (EUROSTAT)

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

Topic (i): New methods for protection of tabular data or for other types of results from table and analysis servers

Re-development of the Cell Suppression Methodology at the US Census Bureau

Prepared by Philip Steel, James Fagan, Paul Massell, Richard Moore Jr., John Slanta, Bei Wang, U.S. Census Bureau, United States of America

Re-development of the Cell Suppression Methodology at the US Census Bureau[1]

Philip Steel, James Fagan, Paul Massell, Richard Moore Jr., John Slanta, Bei Wang*

* This paper represents the joint work of the Research and Methodology team on cell suppression at the United States Census Bureau, Philip Steel is the contact author: philip.m.steel@census.gov.

Abstract: The cell suppression problem is mathematically sophisticated, so the development of the methodology for an application built to solve it poses some interesting design problems. By far the most pressing of the problems is a need for overall speed in producing a solution for a particular set of production tables. This paper presents some lessons learned in building the LP prototype of the modernized cell suppression system for the US Census Bureau.

1 Introduction

This paper presents a collection of ideas that have proven useful in the development of our new cell suppression software. Some may be familiar, but to our knowledge have not been set down elsewhere, perhaps because they fall in between the theory on the subject and papers on specific applications. We will begin with some background on the application: a description of the system and the Operations Research (OR) strategy we employ. Following that, we present a couple of sections on how we organize and process our data. The next section describes our model and the modifications we have made to it. Finally, we explore the identification of already protected cells and the impact it has on our cell suppression and auditing.

The cell suppression re-development project was prompted by aging software and coincident with the impending retirement of the last of the people who developed and ran that system. One of the goals for the methodology group was to devise a Linear Programing (LP) system that could handle extremely large, sparse tables in a reasonable amount of time. Tau Argus was briefly a contender for the basis of the new system, but it is not set up for our batch, network server processing (and its security structure) and, at the time, not open source. We proceeded more or less from scratch, developing a prototype research tool then a specification for the production software, which is currently in its first version. The ideas presented here are part of a three-year effort by our cell suppression Research and Methodology

[1] This article reports the results of research and analysis undertaken by U.S. Census Bureau staff. It has undergone a more limited review than official U.S. Census Bureau publications. It is released to inform interested parties of research and to encourage discussion. The views expressed on statistical, methodological, technical, or operational issues are those of the authors and not necessarily those of the U.S. Census Bureau.

group and our programming staff, most notably Carol Blatt and Vitoon Harusadangkul.

The main methodological difference between the old system and the new system is the adoption of LP rather than network flow. The LP solution avoids some of the problems associated with network: it is valid for higher dimensional data and can accommodate linked tables without losing its formal guarantees for the safeness of the solution. The research goal was to avoid backtracking, which was a bookkeeping exercise to extend the network solution to higher dimensions and across linked tables. Backtracking did not have a formal guarantee of a valid solution and created very complex and error prone processing. The LP methodology does have scaling problems and the US Economic Census tables are among the largest to which cell suppression is applied.

2 OR strategy and company protection

The system is set up in three stages. The purpose of the first stage, which we call the base pass, is to find a protection pattern that guarantees that suppressed cells cannot be estimated within *p* percent of the actual value by table-non-participants. We use a symmetric protection interval (not sliding protection). We set the capacity, the upper bound on flow, to be the cell value and address protection problems arising from companies contributing to more than one cell at a later stage. The first stage applies the solution strategy to a queue of all the designated primary suppressions (Ps). After all the primaries are processed, the second stage examines the resulting pattern for aggregate protection problems and creates a new processing queue of "supercells". That is, we are looking to protect aggregates where the union of suppressed cells fails the p-percent rule because the selected complementary cells (Cs) from the first stage have contribution from one or both of the top two companies of the primary. In the third stage, the model is modified to accommodate the supercells and the solution strategy is applied to each supercell with an adjustment of capacity to reflect a potential suppression's ability to protect the supercell. These processes together produce a protected table.

The solution strategy is inherited from the old system. The LP problem is set up, optimized to minimize the cost times value, then re-optimized with a different cost function over the first solution. The initial cost is weighted value where already suppressed cells have weight 0. The second round cost is 1/(1+value) and the flow in unsuppressed first round cells is bound to 0.

The old system excelled in manipulating the costs in the first optimization, it included a weighting system that was tailored to analysts evaluation of what was important in the tables being produced. We have retained and allowed for elaboration of that weighting. The old system also included a recalculation (on the fly) of capacity, the upper bound on how much "flow" a potential complement could

carry. This was an accounting of company protection specific to each primary. We have deferred implementation of the capacity modification in the base pass due to concerns over speed and rely solely on stage 3 to apply company protection. This may lead to some oversuppression.

3 Linked data and table groups

The system addresses linked data. That is, we want to process data with a model that enforces all the additive relations (a=b+c+d) in the publication. That means that tables that share a margin must be processed together. In terms of the relations: if a pair of relations has a variable in common, then they belong in the same model. We call our processing unit, the cells and the relations for those cells, a table group. We organize our relation files by dimension; we typically have a row relation file, a column relation file and a single level relation.

The following algorithm can be used to determine a dimension in a table group:

> Designate the first relation as belonging to group 1. Take the next relation and determine if it shares any variables with any relations in already designated groups. For the 2nd relation it either shares a variable with the single relation in group 1, in which case it joins the list of members of group1, or it is the first member of group 2. When all the relations in already created groups have been checked for a particular new relation, combine all groups that had a "hit" on that new relation into one group and add the new relation to it. Or, if there were no hits, create a new group. Continue the procedure until the list of relations in that dimension is exhausted.

This procedure is equivalent to constructing the graph where each relation is a node and two nodes have an edge between them if they have at least one shared variable. In the graphical representation, the table groups are the cross product of the maximally connected subgraphs in each dimension. Industry and product coding are hierarchies so their graphs are (connected) trees. Geographic relations are often completely connected as well. In practice, determining table groups usually means divvying up the relations in a single dimension.

If a publication being processed breaks up into distinct table groups, these groups can be run in parallel. The solutions for each are independent.

4 Duplication

One of the idiosyncrasies of our input data is a massive amount of duplication. Part of this arises from a publication convention that extends the detail of some industry

codes. This in turn may be the result of having a data collection system that predates the current coding scheme (NAICS). Multiple geographic schemas also introduce duplication: a unit may appear as a county in one scheme and a metropolitan area "part" in another. Ownership of the unduplication process has been problematic … it doesn't really fit with cell suppression proper. On the other hand, it is desirable to have the prepared cell suppression input data correspond exactly to the publication data-which is duplicated. Fortunately, the unduplication process is compatible with the representation of the data as a graph and we have added it as a pre-process to cell suppression. Duplication enters the relationship files as a simple global equality.

We spent some time looking at local duplication as well. A local equality can allow a trade between constraints and primary suppressions. Consider a parent total that is equal to a child (eg all textile manufacturing in State A is in county B). If the cells are identical in consisting of the report of one company then they are both primaries. One can add a local equality constraint and drop one of the primaries from the processing queue. It may be possible to keep only one cell in the representation of the problem, but this is more complicated for local duplicates than it is for global duplicates. The procedure described in section 8 made this unnecessary.

5 The Evolution of the Model

Unduplication of the data was one step toward a parsimonious statement of the problem. The bulk of that task is in the model itself. Our input typically consists of the non-zero cells, the row relationships, the column relationships and a level relation in one table group. What constraints are needed? What cells need to be represented?

The LP model in the theoretical literature assumes full data. That is, every row, column and level relation generates meaningful constraints, everywhere. In practice, at least for our Economic Census data sets, the data are far from full. It has long been our practice in general data processing to omit unpublished and 0 cells. If we assume the missing data are 0, then following the textbook model produces an excess of constraints, primarily of the form x=0+…+0 or 0=0+…+0. The former corresponds to publications that have data on crossed marginals, but have no interior cells. The latter corresponds to cells in the Cartesian product of the variables where no data were collected. This occurs frequently in economic data that include geographic components with little or no industry of a particular type. The network flow software ignored these empty relations; we have informally called them and their associated cells, whitespace. We are including whitespace formally in the model by its complement, which we are calling A (active). The performance of the software has improved significantly by making the model as lean as possible. We accomplished this in stages.

The 1st model implemented an idea of white space, by cycling through subtables (2 relations and a Cartesian variable in the 3d context) ignoring in the model those

relations that had missing cells. This initial take on the idea required implied cells to be constructed for the input. These cells were predominately 0s.

The one of the exceptions to the missing-data-is-zero rule is worth mention. One of our test datasets uses a fixed base difference estimator. The tables present a total, which has been adjusted to lower variance, along with the unadjusted components. Unfortunately, one of the sources for this adjustment was sensitive administrative data, so despite the general lack of additivity, it was sometimes necessary to include instances of this relation in order to facilitate the protection of sensitive cells. We addressed this in the model by keeping only relations that might lead to a violation of the p% rule. This is accomplished by introducing a valid relation indicator function into the model.

This 1st model produced duplicate constraints … wherever table linkages occurred. While this should simply disappear in the LP pre-solve, but we found that we got another performance boost by eliminating that duplication. It reduced the size of the initial constraint matrix, which may have a ripple effect on the size of objects within the solver. It also seemed to produce a better starting point for the main solve. For the solve improvement we have only anecdotal evidence: we saw reduction in the number of dual simplex iterations (still arriving at the same solution) for problems in the data we were testing at the time. The 2nd model avoided the duplication by looking at each relation and all possible instances of it, applying the same filter. This 2nd model is cleaner; we are looping through all the relations rather than through sub-tables.

The calculation of implied cells was complicated, tedious and inflated the size of the input data. We adopted an alternative method, which ignored 0s and cells that do not belong to any active semi-additive relation. This is reasonable, since zero cells cannot accommodate flow. Likewise only instances of relations that are nominally additive, i.e. where the right hand side and the left hand side are within p% (and not 0), were included as constraints. This 3rd model is presented in 5.1.

The final stage of the model is in the application. The data structure used is a layered graph where the nodes are (nonzero) cells and edges go from totals to summands. No filter is required in this representation, since the active constraints are identical to grouped edges.

5.1 The Model

The model presented is for a three dimensional table, with the notation in full detail. The constraints are grouped by type: (a), (b) and (c) are level, row and column additivity constraints respectively, (d) are the bounds (capacity) and (e) is the target primary's additional equality constraints. (a), (b) and (c) include a screening

condition for degenerate relations (e.g. levs>1). This can be extend to an arbitrary number of dimensions with the notation becoming increasingly difficult or redacted.

$$\textit{minimize: } Y = \sum_{i=1}^{rows} \sum_{\substack{j=1 \\ (i,j,k)\in A}}^{cols} \sum_{k=1}^{levs} c_{i,j,k} \left(x_{i,j,k}^{(u)} + x_{i,j,k}^{(l)} \right)$$

subject to:

(a) $$\sum_{\substack{k=2 \\ (i,j,k)\in A}}^{levs} \left(x_{i,j,k}^{(u)} - x_{i,j,k}^{(l)} \right) = x_{i,j,1}^{(u)} - x_{i,j,1}^{(l)}$$

for i =1, ... , rows, j = 1, ... ,cols : *levs* > 1, $ws(i,j,1) = 0$

(b) $$\sum_{\substack{i=1 \\ (i,j,k)\in A}}^{\lim r(ii)} \left(x_{rowrel(ii,i),j,k}^{(u)} - x_{rowrel(ii,i),j,k}^{(l)} \right) = x_{rowrel(ii,0),j,k}^{(u)} - x_{rowrel(ii,0),j,k}^{(l)}$$

for ii = 1, ... , rr, j = 1,..,cols, k = 1, ... , *levs* : $limr(ii) \geq 1$, $ws(ii,j,k) = 0$

(c) $$\sum_{\substack{j=1 \\ (i,j,k)\in A}}^{\lim c(jj)} \left(x_{i,colrel(jj,j),k}^{(u)} - x_{i,colrel(jj,j),k}^{(l)} \right) = x_{i,colrel(jj,0),k}^{(u)} - x_{i,colrel(jj,0),k}^{(l)}$$

for i = 1, ... , *rows*, jj = 1, ... , cc, k = 1, ... , *levs* : $limc(cc) \geq 1$, $ws(i,jj,k) = 0$

(d) $$0 \leq x_{i,j,k}^{(u)} \leq h_{i,j,k} \;;\; 0 \leq x_{i,j,k}^{(l)} \leq h_{i,j,k}$$

for i = 1, ... , *rows*, j = 1, ... , *col*, k = 1, ... , *levs* : $(i,j,k) \in A$

(e) $$x_{prow,pcol,plev}^{(u)} = prot \;;\; x_{prow,pcol,plev}^{(l)} = 0$$

where:

$$c_{i,j,k} = \begin{cases} \max(0, v_{i,j,k}) & when\ (i,j,k) \in U \\ 0 & when (i,j,k) \in P \cup C \end{cases}$$

$$h_{i,j,k} = \max(0, v_{i,j,k})$$

rows = number of rows of the 3d table
cols = number of columns of the 3d table
levs = number of levels of the 3d table
rr = number of row relations
cc = number of column relations

limr(i) = number of summands in row relation i, i = 1, ... , rr
limc(j) = number of summands in column relation j, j = 1, ... , cc
rowrel(ii,i) = ith row number in the iith row relation, ii = 1, ... , rr : i, i = 1, ... , limr(ii)

rowrel(ii,0) = row number of the marginal in the iith row relation, ii = 1, ... , rr :
colrel(jj,j) = jth column number in the jjth column relation, jj = 1, ... , cc : j, j = 1, ... , limr(jj)
colrel(jj,0) = column number of the marginal in the jjth row relation, jj = 1, ... , cc

prow = row number of the target primary cell
pcol = column number of the target primary cell
plev = level number of the target primary cell
prot = protection requirement of the target primary cell

$h_{i,j,k}$ = capacity of cell (i,j,k)
$v_{i,j,k}$= value of cell (i,j,k)
$c_{i,j,k}$ = cost coefficient of cell (i,j,k)

A = the set of active cells in the Cartesian table space:
{(i,j,k) s.t. $v_{i,j,k}>0$ and ws(i,j,1)=0, ws(i,jj,k)=0 or ws(ii,j,k)=0 for some ii,jj (&1)}
U = set of unsuppressed cells (with a value) in the table
P = set of primary cells in the table
C = set of complementary cells in the table

ws(i,j,1) indicates whether the level relation is valid for the ith row and jth column:

$$\text{ws(i,j,1)=0 if } \frac{abs\left(v_{i,j,1} - \sum_{k=2}^{levs} v_{i,j,k}\right)}{v_{i,j,1}} < p \text{ and } v_{i,j,1}>0 \text{ and } \sum_{k=2}^{levs} v_{i,j,k} > 0;$$

ws(i,j,1)=1 otherwise.

ws(i,jj,k) indicates whether the jjth column relation is valid for the ith row and kth level:

$$\text{ws(i,jj,k) = 0 if } \frac{abs\left(v_{i,1,k} - \sum_{j=2}^{\lim c(jj)} v_{i,colrel(jj,j),k}\right)}{v_{i,1,k}} < p \text{ and } v_{i,1,k}>0 \text{ and } \sum_{j=2}^{\lim c(jj)} v_{i,colrel(jj,j),k} > 0$$

ws(i,jj,k)=1 otherwise.

ws(ii,j,k) indicates whether the iith row relation is valid for the jth column and kth level:

$$\text{ws(ii,j,k) = 0 if } \frac{abs\left(v_{1,j,k} - \sum_{i=2}^{\lim r(ii)} v_{rowrel(ii,i),j,k}\right)}{v_{1,j,k}} < p \text{ and } v_{1,j,k}>0 \text{ and } \sum_{i=2}^{\lim r(ii)} v_{rowrel(ii,i),j,k} > 0$$

ws(ii,j,k)=1 otherwise.

Note that we are treating negative value data as 0 (no flow variables) except to qualify the relation. This may cause an infeasibility. If the negative value requires protection it must be done outside the system.

6 Software and hardware

The prototype is an amalgam of SAS, FORTRAN, AMPL and CPLEX. The production version is c++ interacting with CPLEX through a graph object using the Boost c++ library. They are run in a red hat linux system with 8, 2.93 GHz Intel Xeon processors and 60GB memory.

7 Depth

We order our processing in the computationally intensive first stage by size of protection requirement within depth. We have several implementations of depth, but all are approximations of the number of steps from the cell to the closest grandtotal. We do not make any claim that this consistently produces a best order; however, this top down order did better than the other orders we tried. We reuse this measure to establish a rank among discounted cells.

8 Skip P

When we scaled our prototype up to a work on a medium sized table, we started experiencing some rather daunting processing times. This table was very sparse, with almost half the cells marked as primaries. It was evident that there were very few publishable cells in the deep interior of the table group. However it was also evident that most P were already being protected by virtue of the other suppressions present … the process log showed 95% of the problem objectives were coming back 0 (already protected). We spent several fruitless weeks trying to find criteria to exclude these P from the processing queue. Then we had the idea of checking a solution in hand to see if it protected P other than the target P. We had the good fortune to be working on another problem at about the same time. We were looking at the trace log to try to create a representation of the initial protection patterns and ran into a problem: we were unable to sort out the cost 0 cells (Ps & Cs) actually involved in protecting the target P. In addition to having too many cells, the flow in the cost 0 cells was inflated--more than what was needed to protect the target primary. This made it impossible (without another LP) to figure out what flow was actually needed in those cells to protect the target. However, these same properties, which prevent a peak at the local protection pattern, are just what is needed for finding P, other than the target, which are protected by the overall pattern in its current state.

Whenever we have a solution in hand, where we are committed to retaining the suppressions, we check the flow in each of the P cells that remain in the processing queue. If these cells have flow greater than their protection need, we mark them to be skipped.

8.1 Performance on sector 71 data

Disclosure Group 7 is a geographic unit consisting of the NE states; this includes all of the pieces of the Boston metropolitan area. The sector 71 data has a relatively small NAICS hierarchy. The columns are geographic units that are linked and non-hierarchical. The third dimension is simple: the total is the sum of taxable and non-taxable company receipts. These data are relatively small for publications in our Economic Census' Geographic Area Series. The initial input consists of 42,198 cells, which reduces to 17,467 after unduplication. 11,037 of those are primary suppressions. The unduplicated Cartesian space it sits in has 88,908 cells. Skip P allowed us to process just 588 of the 11,037 P. The prototype took 25 minutes; the production LP took 3.5 minutes. The LP methodology gives a solution with 15% fewer C and publishing around 20% more priority value, compared to the old network flow.

8.2 Proof of Skip P

Note that the model (see section 5.1) for protecting P1 and the model for protecting P2 are the same except for (e), the last two constraints. So a solution (X^u, X^d) for P1 can be transformed to a solution for P2 by applying the scalar prot2/max(x^u_{P2},x^d_{P2}) where prot2 is the protection requirement for P2 and the maximum is the "flow" through P2 in P1's solution, **provided that the bounding constraints (d) are still satisfied**. This is guaranteed if the scalar is less than 1, ie max(x^u_{P2},x^d_{P2}) >=prot2.

8.3 The fat solution

The somewhat troubling aspect of Skip P is that we are uncertain why the solver is providing such a fat solution. By "fat solution" we mean that the cost 0 domain (Ps and Cs) is in full play, almost every point that *can* support flow under the equality constraints *does*. In addition, the amount of flow is close to its maximum regardless of the protection requirement of the P being processed!

The counts from the optimization log of our sector 71 data are instructive (figure 1). This table had 11037 P. The first completed optimization (after inverted cost) had a solution with flow in 3961 P, 3076 of which passed the Skip P test. I.e., we solved 3077 problems with that first optimization. Note that in the last optimization **all** P had flow and that we have managed to skip 10,449 P (95%) over the whole table group.

The final problem lights up the entire solution with flow; this may be approximately the collection of maximal symmetric intervals that do not exceed the bounds on the cell (as determined by a full audit). We speculate that the implementation of the dual simplex--and perhaps other techniques as well--favours finding the fat domain (cost 0 part) of the solution first, since all solutions being compared in the algorithm are

being constrained by the flow through the fat domain. This reduces the calculations necessary to compare solutions.
The single set of maximum flow values in the interior of the fat domain will fulfil the equality constraints for **any** valid solution (minimal or otherwise) on the cost-greater-than-0 domain, with minor modification of boundary values immediately touching "new" C.

The impact of the Skip P procedure is almost certainly going to be far less on tables that lack the sparseness exhibited by our Geographic Area Series publication. Even so, there are very few things in life that are free--you need to take advantage of them when they come along.

Optimization number	Count of P with flow	Running total of skipped P
1	3961	3076
2	3952	3243
.	.	.
.	.	.
.	.	.
587	11035	10448
588	11037	10449

Figure 1 Effectiveness of the Skip P procedure

9 Quickaudit

One application of the Skip P idea is to auditing. LP is self-auditing in the sense that if the solver does not come across an infeasible problem, the solution does not undersuppress provided the model is correct. However, if a table is protected by some other methodology, then the bounds problem under the LP model can determine undersuppression. With Skip P we were able to audit network flow solutions on very large tables that were previously "un-auditable". That is, in a 5 minute run on one of our network solutions we were able find protection problems that would have otherwise taken weeks of processing time to identify with a conventional audit. Quickaudit works by selecting a small set of cells, inputting the solution to be audited, solving the protection problem for each of the small set of cells … if the objective function is 0 (as the solution represents) for a cell then the Skip P procedure can identify cells whose bounds are beyond p%. Since the solution is supposed to be complete, we are at the stage of the overall problem where the solutions are at their fattest. So Skip P can identify most of the protected cells. We then can apply a conventional audit to the remainder.

10 Conclusion and Further Research

We have produced a methodology/application that allows for the protection of large tables in a reasonable amount of time, with considerably less over-suppression than the network flow system we are replacing. The project required a mix of application experience, theoretical knowledge and programming skills. We have described the methodological enhancements that helped reduce the processing time. The actual production software, which does a much better job in handling the data and interacting with the solver, beats our research prototype by a factor of 5--even on the most computationally intensive files.

We plan to research the effectiveness and processing cost of including cost adjustments for company protection in the base pass. Wang (2013) explores the trade-off between m-LP time savings and over-suppression. This is one way to protect files beyond the range of the new production software and is an easy adaptation of that system. We also plan to explore an extension of global un-duplication to local cell duplicates, to add another efficiency to our data representation.

We need a better understanding of the fat solution described in section 8.3, in particular whether it can be guaranteed for certain solve techniques and/or software. We also would like to develop some heuristics for weighting data in a manner that improves the utility of the published data.

References

Federal Committee on Statistical Methodology (1994) *Statistical Policy Working Paper 22 (Revised 2005)-Report on Statistical Disclosure Limitation Methodology* NTIS PB94-165305.

Jewett, R. *Disclosure Analysis for the 1992 Economic Census*: unpublished US Census Bureau documentation 1993.

Kirkendahl, N. and Sande, G. *Comparison of Systems implementing Automated Cell Suppression for Economic Statistics*. Journal of Official Statistics, Vol. 14, No. 4, 1998 pp. 513-535.

Massell, P. B. *Using Linear Programming for Cell Suppression in Statistical Tables: Theory and Practice*. Proceedings of the Annual Meeting of the American Statistical Association, August 5-9, 2001.

Giessing, S. *A Survey on Software Packages for Automated Secondary Cell Suppression*. Joint ECE/Eurostat Work Session on Statistical Data Confidentiality, March 1999.

Giessing, S. *Protection of tables with negative values*. CASC report on the ESSNet SDC website, (ca 2010).

Wang, B. *Improving LP performance in Cell Suppression Process*. Proceedings of the Annual Meeting of the American Statistical Association, August 3-8, 2013

Hundepool, A, et al. *Tau Argus User's Manual* 2008.

Tambay, J-L. et al., *Treatment of Aggregated Sensitive Cells*. Statistics Canada working paper, 2007.