Modeling and Solving the Cell Suppression Problem for Linearly-Constrained Tabular Data

Matteo Fischetti[1] & Juan José Salazar[2]

[1] DEI, University of Padova, Italy; fisch@dei.unipd.it
[2] DEIOC, University of La Laguna, Spain; jjsalaza@ull.es

Abstract. We study the problem of protecting sensitive data in a statistical table whose entries are subject to a system of linear constraints. This very general setting covers, among others, k-dimensional tables with marginals as well as linked tables. In particular, we address the $\mathcal{NP}$-hard problem known in the literature as the (secondary) Cell Suppression Problem. We introduce a new integer linear programming model and describe additional inequalities used to strengthen the linear relaxation of the model. We also outline a branch-and-cut algorithm for the exact solution of the problem, which can also be used as a heuristic procedure to find near-optimal solutions. Preliminary computational results are promising.

Keywords: Statistical Disclosure Control, Cell Suppression, Integer Linear Programming.

1 Introduction

A statistical agency collects data to be processed and published. Raw material is information obtained from individual respondents. Usually, this data is obtained under a pledge of confidentiality: statistical agencies have the responsibility of not releasing any data or data summaries from which individual respondent information can be revealed (*sensitive data*). On the other hand, statistical agencies aim at publishing as much information as possible. This results in a trade-off between privacy rights and information loss, an issue of primary importance in practice. We refer the interested reader to Willenborg and Waal [15] for an in-depth analysis of statistical disclosure control methodologies.

Starting in 1996, the European Union supported through EUROSTAT (the European statistical office) a 3-year ESPRIT research project aimed at developing and testing new methodologies within statistical disclosure control. The project, coordinated by Dr. Leon Willenborg from CBS (Central Bureau of Statistics, The Netherlands), involves several research groups from both academia and national statistical offices. We participate in the project for the definition of mathematical models and solution algorithms for protecting sensitive information in tabular data.

	A	B	C	Total
Activity I	20	50	10	80
Activity II	8	19	22	49
Activity III	17	32	12	61
Total	45	101	44	190

(a) Original table

	A	B	C	Total
Activity I	20	50	10	80
Activity II	*	19	*	49
Activity III	*	32	*	61
Total	45	101	44	190

(b) Published table

Fig. 1. Investment of enterprises by activity and region.

Cell suppression is one of the most popular techniques for disclosure avoidance, and is typically applied to 2- or 3-dimensional tables whose entries (*cells*) are subject to marginal totals. In this paper we address cell suppression in the more general case in which data is subject to a generic system of linear constraints. Hence our study covers, among others, k-dimensional tables with marginals as well as linked tables. To our knowledge, this is the first attempt to model and solve the cell suppression problem in such a very general context.

We will introduce the basic cell suppression problem with the help of a simple 2-dimensional example, taken from Willenborg and Waal [15]. Figure 1(a) exhibits a table giving the investment of enterprises (per millions of guilders), classified by activity and region. Let us assume that the information in the cell corresponding to Activity II and Region C is confidential, hence it is viewed as a *sensitive cell* to be suppressed. But that is not enough: by using the marginal totals, an *attacker* interested in the disclosure of the sensitive cell can easily recompute its missing value. Then other table entries must be also suppressed; see Figure 1(b). With this choice, the attacker cannot disclose the nominal value of sensitive cell exactly, although he can still compute a range for the values of this cell which are consistent with the published entries. Indeed, the minimum value $\underline{y}_{23}$ for sensitive cell can be computed by solving a linear program in which the values y'_{ij} for the missing cells (i,j) are treated as unknowns, namely

$$\underline{y}_{23} := \min y'_{23}$$

subject to

$$\begin{array}{rrrrl} y'_{21} & & +y'_{23} & & = 30 \\ & y'_{31} & & +y'_{33} & = 29 \\ y'_{21} & +y'_{31} & & & = 25 \\ & & y'_{23} & +y'_{33} & = 34 \\ y'_{21}, & y'_{23}, & y'_{31}, & y'_{33} & \geq 0. \end{array}$$

Notice that the right-hand-side values are known to the attacker, as they can be obtained as the difference between the marginal and the published values in a row/column.

The maximum value $\overline{y}_{23}$ for the sensitive cell can be computed in a perfectly analogous way, by solving the linear program $\max y'_{23}$ subject to the same constraints as before.

In the example, $\underline{y}_{23} = 5$ and $\overline{y}_{23} = 30$, i.e., the sensitive information is "protected" within the *protection interval* [5,30]. If this interval is considered sufficiently wide by the statistical office, then the sensitive cell is called *protected*, otherwise new suppressions are needed. Notice that the protection interval $[5, 30]$ is only attained if the cell corresponding to Activity II and Region A takes the quite unreasonable values of 0 and 25. Bounding the cell variation to $+/-50\%$ of the nominal value (say) results in the more realistic protection interval $[18, 26]$. This example illustrates the need of considering "external bounds" on the cell values when computing realistic protection intervals.

A more detailed description of the problem is as follows. Given a set of sensitive cells PS (*primary suppressions*) along with the required protection levels, the statistical office aims at finding a set of secondary suppressions protecting all the sensitive cells against the attacker, and such that the loss of information associated with the suppressed entries is minimized.

More specifically, let $T = 1, \ldots, n$ be the index set of a given table to be protected. The values y_i of the entries of T are subject to a given set of linear equations, say $Ay = b$ (our model can easily be extended to the case of linear inequalities). Moreover, each entry i of the table has an associated *weight* $w_i \geq 0$ giving a measure of the loss of information incurred if the entry is suppressed. We assume that the attacker knows a range of feasible values ("external bounds") for each table entry a_i, say $[a_i - lb_i, a_i + ub_i]$.

Let $i_1, \ldots, i_p$ be the $p := |PS|$ sensitive cells in PS. Each sensitive cell i_k is considered *protected* when the associated protection interval $[\underline{y}_{i_k}, \overline{y}_{i_k}]$ computed by a possible attacker contains the closed segment $[a_{i_k} - LPL_k, a_{i_k} + UPL_k]$, where the non-negative *lower* and *upper protection levels* LPL_k and UPL_k are provided by the statistical office. The problem then calls for a minimum-weight set of cells whose suppression protects all sensitive cells. This combinatorial optimization problem is known as the (secondary) *Cell Suppression Problem* (CSP), and belongs to the class of the strongly $\mathcal{NP}$-hard problems; see, e.g., Kelly, Golden and Assad [10], Geurts [6], Kao [8].

Previous works on CSP mainly concentrate on heuristics for 2-dimensional tables with marginals; see, e.g., Cox [1, 2], Sande [14], Kelly, Golden and Assad [10], and Carvalho, Dellaert and Osório [4]. Robertson [13] extended the algorithm of Sande [14] to 3-dimensional tables.

Kelly [9] proposed a mixed-integer linear programming formulation involving a huge number of variables and constraints. Geurts [6] refined this model, and reported computational experiences on small-size instances, the largest instance solved to optimality being a table with 20 rows, 6 columns and 17 sensitive cells. Gusfield [7] gave a polynomial algorithm for a special case of the problem. Fischetti and Salazar [5] presented a new method capable of solving to proven optimality instances with up to 250,000 cells and 10,000 sensitive cells on a personal computer; a preliminary version of [5] was presented by the authors at the *3rd International Seminar on Confidentiality* held in Bled (Slovenia) in 1996.

2 An integer linear programming model

As already mentioned, we address the very general case in which table entries y_i are linked by a generic system of linear equations $Ay = b$.

In the case of k-dimensional tables with marginals, A is a $\{0, +1, -1\}$-matrix and $b = 0$. Moreover, for the case $k = 2$ the linear system can be represented in a natural way as a *network*; see, e.g., Cox [1, 2] and Kelly, Golden and Assad [10]. Unfortunately, this nice structure is not preserved for $k \geq 3$, unless the table decomposes into a set of independent 2-dimensional subtables.

We say that a vector $[y_i]$ defines a *feasible table* whenever $Ay = b$, and $a_i - lb_i \leq y_i \leq a_i + ub_i$ holds for all i. Clearly the original table $[a_i]$ defines a feasible table, that we call the *nominal table.*

Given a set SUP of suppressed cells, the *attacker problem* consists of computing the minimum/maximum values for each cell of PS in a feasible table which coincides with the nominal one for the published cells (those not belonging to SUP). This problem splits into the following subproblems, each associated with a different $i_k \in PS$:

$$\left.\begin{array}{rl} \underline{y}_{i_k} := & \min y'_{i_k} \\ & Ay' = b \\ & a_i - lb_i \leq y'_i \leq a_i + ub_i \text{ for } i = 1, \ldots, n \\ & y'_i = a_i \qquad \text{for all } i \notin SUP \end{array}\right\} \tag{1}$$

$$\left.\begin{array}{rl} \overline{y}_{i_k} := & \max y''_{i_k} \\ & Ay'' = b \\ & a_i - lb_i \leq y''_i \leq a_i + ub_i \text{ for } i = 1, \ldots, n \\ & y''_i = a_i \qquad \text{for all } i \notin SUP \end{array}\right\} \tag{2}$$

where the continuous variables y' and y'' are "local" to each subproblem.

In order to formulate the overall CSP as a Mixed-Integer Linear Programming (MILP) model we introduce a binary variable $x_i = 1$ for each table entry i, where $x_i = 1$ if $i \in SUP$ (suppressed), and $x_i = 0$ otherwise (published). We obtain:

$$\min \sum_{i=1}^{n} w_i x_i$$

subject to $x \in \{0, 1\}^n$ and, for each $i_k \in PS$:

$$x_{i_k} = 1,$$

$$\left.\begin{array}{rl} LPL_k - a_{i_k} \leq -\underline{y}_{i_k} = & \max\,(-y'_{i_k}) \\ & Ay' = b \\ & a_i - lb_i x_i \leq y'_i \leq a_i + ub_i x_i \text{ for } i = 1, \ldots, n \end{array}\right\} \tag{3}$$

$$\left.\begin{array}{rl} UPL_k + a_{i_k} \leq \overline{y}_{i_k} = & \max y''_{i_k} \\ & Ay'' = b \\ & a_i - lb_i x_i \leq y''_i \leq a_i + ub_i x_i \text{ for } i = 1, \ldots, n \end{array}\right\} \tag{4}$$

Standard MILP solution techniques such as branch-and-bound or cutting-plane methods require the solution of the Linear Programming (LP) relaxation of the model in hand, obtained by relaxing conditions $x_i \in \{0,1\}$ into $0 \leq x_i \leq 1$ for all i. However, even the LP relaxation of our model is very difficult to solve, in that it involves a really huge number of "local variables" y' and y'' and of "linking contraints" between the x and the y variables.

On the other hand, the rich arsenal of Mathematical Programming tools for solving hard MILP's include a technique which appears very suited for our case, namely the Benders' approach described, e.g., in Nemhauser and Wolsey [11, Section II.3.7].

The method was originally proposed by Benders in 1962 as a decomposition technique for solving large-scale MILP's. We use the basic idea of the method within a modern branch-and-cut framework, thus overcoming some of the drawbacks that discouraged its use in the past. Here is a sketch of the approach we use.

We want to solve the LP relaxation of our model without introducing explicitly the local variables y' and y'' and the associated linking constraints. In other words, we always work in the space of x variables only.

We start by solving the so-called *master problem*, which is initially defined as the trivial LP:

$$\min\{\sum_{i=1}^{n} w_i x_i : x_{i_1} = \ldots = x_{i_p} = 1,\ x \in [0,1]^n\}.$$

Let x^* be the optimal solution found. Our order of business it to check whether the vector x^* allows for suitable local vectors y' and y'' which satisfy the missing constraints (3)-(4) when x is fixed to the given x^*. To this end, we invoke a "verification procedure" which tries to compute such a local vectors. As x^* is given, this task can be performed efficiently as it amounts to the solution of a sequence $2p$ small LP's, each associated with a single lower/upper protection level. More specifically, for each $i_k \in PS$ the verification procedure has to solve the LP (3) with $x = x^*$ to check the fulfillment of the lower protection level, and the LP (4) to check the fulfillment of the upper protection level.

If successful, the verification procedure returns the required local vectors which certify the feasibility and hence the optimality of x^*. If this is not the case, at least one local LP found a maximum value strictly less than the imposed limit $LPL_k - a_{i_k}$ or $UPL_k + a_{i_k}$. For each such LP, the optimal LP dual solution allows one to immediately derive a valid linear inequality —in the x variables only— which is violated by x^*. This inequality is called *Benders' cut*, and expresses a property shared by all the feasible vectors x (those which can be completed by suitable local vectors) but not by the given x^*, thus certifying the infeasibility of x^*.

The Benders' cuts obtained in this way are then appended to the current master LP, which is reoptimized through parametric techniques (e.g., the dual simplex method) to identify a new trial x^* on which the procedure is iterated.

After a (usually short) number of iterations, the above scheme produces the required optimal solution x^* of the LP relaxation of our model. If x^* is integer, then it corresponds to an optimal solution of the original MILP, and we are done. Otherwise, an enumerative scheme akin to branch-and-bound is applied to force the integrality of x^*. During the enumertion, new Benders' cuts are identified on the fly and added to the corresponding master LP's. This produces an overall solution scheme called *branch-and-cut* [12], which merges run-time cut generation and branching into a very flexible framework.

It is worth noting that the above approach differs from the text-book one in a critical respect: we generate Benders' cuts for each (possibly fractional) solution of the current master LP, whereas in the original proposal of Benders such cuts are only generated when the enumeration produces an integer solution. In this way we avoid wasting computing time on branching on fractional solutions x^* which can be cut off by means of Benders' cuts.

In our context, Benders' cuts are of the form

$$\left.\begin{aligned} \textstyle\sum_{i=1}^n (\beta_i' ub_i - \alpha_i' lb_i) x_i \geq LPL_k, \\ \textstyle\sum_{i=1}^n (\beta_i'' ub_i - \alpha_i'' lb_i) x_i \geq UPL_k, \end{aligned}\right\} \tag{5}$$

where $(\alpha', \beta', \gamma')$ and $(\alpha'', \beta'', \gamma'')$ are (optimal) dual solutions of the attacker subproblems (3) and (4), respectively. We call these cuts *capacity constraints* in analogy with similar constraints we introduced in [5] for the network interpretation associated with 2-dimensional tables.

Each capacity cut constraint can be strengthened by replacing $(\beta_i' ub_i - \alpha_i' lb_i)$ by $\min\{\beta_i' ub_i - \alpha_i' lb_i, LPL_k\}$, and $(\beta_i'' ub_i - \alpha_i'' lb_i)$ by $\min\{\beta_i'' ub_i - \alpha_l'' lb_i, UPL_k\}$, These simple improvements are very effective in practice, and typically reduce considerably the number of branchings required to find the optimal solution.

3 Additional inequalities

Additional inequalities can be used for strengthening the LP relaxation of our model. We propose two main classes of additional inequalities, that we call *bridge* and *cover* inequalities. Computational results show that these inequalities allow for a considerable speed-up in the problem resolution.

3.1 Bridge inequalities

As the weights w_i are assumed to be nonnegative, every CSP instance has an optimal solution in which no suppression is redundant. Therefore, one can require that the value of each entry $i \notin PS$ with $x_i = 1$ cannot be recomputed exactly. This is equivalent to associate a very small fictitious "sliding protection level" $\epsilon > 0$ to each suppressd cell not in PS, and to set-up the associated attacker subproblem with the requirement that

$$\overline{y}_i - \underline{y}_i \geq \epsilon x_i.$$

By means of a technique similar to that we used to derive capacity constraints, one can obtain the following *bridgeless inequalities*:

$$\sum_{l:(\beta'_l+\beta''_l)ub_l>(\alpha'_l+\alpha''_l)lb_l} x_l \geq x_i,$$

where $(\alpha', \beta', \gamma')$ and $(\alpha'', \beta'', \gamma'')$ are dual solutions of properly-defined (fictitious) attacker subproblems associated to cell i.

3.2 Cover inequalities

Capacity cuts have been derived without using the fact that x-variables must be 0-1 valued, a property that can be exploited to derive strong families of additional constraints which are useful to tighten the LP relaxation of our model.

In particular, following the seminal work of Crowder, Johnson and Padberg [3] we can observe that each single capacity inequality can be viewed as a k-napsack constraint, hence it implies a number of "more combinatorial" restrictions. To be specific, let $\sum_i s_i x_i \geq s_0$ represent any inequality in (5), in the strengthened form discussed at the end of Section 2. By means of the variable transformation $x_i = 1 - \xi_i$ we can bring the inequality to the form

$$\sum_i s_i \xi_i \leq \sum_i s_i - s_0.$$

Along with the 0-1 condition on the ξ-variables, this constraint defines a k-napsack problem with item weights s_i's and capacity $W := \sum_i s_i - s_0$. Hence any constraint that is valid for this knapsack problem is also valid for CSP. In particular, we have considered the well-known *cover inequalities*:

$$\sum_{i \in Q} \xi_i \leq |Q| - 1, \text{ for each cell subset } Q : \sum_{i \in Q} s_i > W = \sum_i s_i - s_0, \tag{6}$$

which forbid the choice of all the items in any set Q whose global weight exceeds the knapsack capacity. In terms of the original variables, these inequalities read

$$\sum_{i \in Q} x_i \geq 1, \text{ for each cell subset } Q : \sum_{i \notin Q} s_i < s_0, \tag{7}$$

and require the suppression of at least one element of certain sets Q of cells.

4 A branch-and-cut algorithm

The model presented in the previous sections contains a huge number of constraints. As already mentioned, we address its solution by using a *branch-and-cut* technique, as introduced in Padberg and Rinaldi [12].

The approach can be viewed as an enumerative scheme akin to branch-and-bound, in which at every node of the branching tree a reduced LP relaxation of our model (in the x variables only) is solved in order to obtain a lower bound on the optimal solution value. At every branching node we start by solving an LP relaxation with only a small part of the constraints imposed. Then, iteratively, we consider the optimal solution x^* of the current LP relaxation and look for new inequalities which are valid for CSP but violated by x^*. This is called the *separation phase*. If violated inequalities are found, we add (some of) them to the current LP relaxation, which is reoptimized. In order to avoid LP problems with too many constraints we also remove from the current LP some of the inequalities which are slack at x^*. These inequalities are stored in a *constraint pool* data-structure, and re-inserted in the current LP as soon as they become violated by the new LP solutions. The whole procedure is iterated until no violations are found. At this point, if x^* is integer we have an optimal CSP solution of the subproblem associated with the current branching node, and we proceed to elaborate the remaining subproblems (if any) according to a best-bound-first strategy. Otherwise we select a fractional variable x_i and create two new subproblems by fixing x_i to 0 and to 1, respectively (*branching phase*).

A key point of the approach is the availability of effective separation algorithms. We have implemented exact separation procedures, based on linear programming, for the capacity (Benders) cuts and the bridgeless inequalities, as well as a heuristic scheme for cover inequalities.

5 Conclusions

Cell suppression is a widely-used methodology in Statistical Disclosure Control. In this paper we have introduced a new integer linear programming model for the cell suppression problem in the very general context of tables whose entries are subject to a generic system of linear constraints. The model covers k-dimensional tables with marginals as well as linked tables. To our knowledge, this is the first attempt to model and solve the cell suppression problem in such a very general context. We have also outlined a possible solution procedure in the branch-and-cut framework. Preliminary computational results are encouraging.

6 Acknowledgment

Work partially supported by the European Union - ESPRIT project 20462-SDC on Statistical Disclosure Control. We thank Alberto Caprara for his help in code design and testing.

References

1. L.H. Cox, "Suppression Methodology and Statistical Disclosure Control", *Journal of the American Statistical Association*, 75 (1980) 377–385.
2. L.H. Cox, "Network Models for Complementary Cell Suppression", *Journal of the American Statistical Association*, 90 (1995) 1453–1462.
3. H.P. Crowder, E.L. Johnson, and M.W. Padberg, "Solving Large-Scale Zero-One Linear Programming Problems", *Operations Research*, 31 (1983) 803–834.

4. F.D. Carvalho, N.P. Dellaert, M.S. Osório, "Statistical Disclosure in Two-Dimensional Tables: General Tables", *Journal of the American Statistical Association*, 89 (1994) 1547–1557.
5. M. Fischetti, J.J. Salazar, "Models and Algorithms for the 2-Dimensional Cell Suppression Problem in Statistical Disclosure Control", Working Paper, 1997.
6. J. Geurts, "Heuristics for Cell Suppression in Tables", working paper, Netherlands Central Bureau of Statistics, Voorburg, 1992.
7. D. Gusfield, "A Graph Theoretic Approach to Statistical Data Security", *SIAM Journal on Computing* 17 (1988) 552–571.
8. M.Y. Kao, "Data Security Equals Graph Connectivity", *SIAM Journal on Discrete Mathematics*, 9 (1996) 87–100.
9. J.P. Kelly, "Confidentiality Protection in Two and Three-Dimensional Tables", Ph.D. dissertation, University of Maryland, College Park, Maryland, 1990.
10. J.P. Kelly, B.L. Golden, and A.A. Assad, "Cell Suppression: Disclosure Protection for Sensitive Tabular Data", *Networks* 22 (1992) 397–417.
11. G.L. Nemhauser, L.A. Wolsey, "Integer and Combinatorial Optimization, John Wiley & Sons, 1988.
12. M. Padberg, G. Rinaldi, "A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems", *SIAM Reviews*, 33 (1991) 60–100.
13. D.A. Robertson, "Cell Suppression at Statistics Canada", Proceedings of the *Second International Conference on Statistical Confidentiality*, Luxembourg, 1995.
14. G. Sande, "Automated cell suppression to preserve confidentiality of business statistics", *Statistical Journal of the United Nations ECE* 2 (1984) 33–41.
15. L.C.R.J. Willenborg, T. de Waal, "Statistical Disclosure Control in Practice", *Lecture Notes in Statistics 111*, *Springer*, New York, 1996.

This article was processed using the LaTeX macro package with LLNCS style