Journal of Official Statistics, Vol 14, No. 4, 1998, pp. 553–565

Experiments with Controlled Rounding for Statistical Disclosure Control in Tabular Data with Linear Constraints

Matteo Fischetti[1] *and Juan-José Salazar-González*[2]

In this article we describe theoretical models and practical solution techniques for protecting confidentiality in statistical tables containing sensitive information that cannot be disseminated. This is an issue of primary importance in practice. We study the problem of protecting sensitive information in a statistical table whose entries are subject to any system of linear constraints. This very general setting covers, among others, k-dimensional tables with marginals as well as hierarchical and linked tables. In particular, we address the $\mathcal{NP}$-hard optimization problem known in the literature as the (zero-restricted) Controlled Rounding Problem. We also propose a modification of this problem, which allows for enlarged rounding windows in case the zero-restricted version is proved to have no feasible solution. We describe integer Linear Programming (LP) models and introduce effective LP-based enumerative algorithms, which have been embedded within τ-ARGUS, a software package for statistical disclosure control. Computational results on 2-, 3-, and 4-dimensional tables are presented. An interesting outcome is that 4-dimensional tables often admit no zero-restricted rounding, whereas slightly enlarged rounding windows produced feasible instances in all the cases in our test bed.

Key words: Statistical disclosure control; confidentiality; controlled rounding; integer linear programming.

1. Introduction

A statistical agency collects data to be processed and published. 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. 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 De Waal (1996) for an in-depth analysis of statistical disclosure control methodologies.

Controlled rounding is a widely-used technique for disclosure avoidance, and is typically applied to 2- or 3-dimensional tables whose entries (*cells*) are subject to marginal totals; see Fellegi (1972). We will introduce the basic controlled rounding problem with the help of a simple example, taken from Willenborg and De Waal (1996). Figure 1(a) exhibits a 2-dimensional table giving the investment of enterprises (per million of

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

Acknowledgments: The work was partially supported by the European Union through ESPRIT project 20462-SDS on Statistical Disclosure Control. The first author was supported by Ministero della Ricerca Scientifica e Tecnologica, Italy, and the second author was supported by Ministerio de Educación y Ciencia, Spain. We thank Alberto Caprara who implemented the separation procedures for Gomory cuts and for {0, 1/2}-cuts.

	Region			
	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

	Region			
	A	B	C	Total
Activity I	20	50	10	80
Activity II	10	20	20	50
Activity III	15	30	15	60
Total	45	100	45	190

(b) Published (rounded) table

Fig. 1. Investment of enterprises by activity and region

guilders), classified by activity and region. Let us assume that the Statistical Office wants to protect the sensitive information of the table by ''perturbing'' its entries by small amounts. For instance one can consider rounding all the table entries to the nearest integer multiple of 5 (say). However, rounding in this way the entries in the row corresponding to Activity III would lead to the inconsistency $15 + 30 + 10 = 60$ in the marginal sum. To avoid this drawback, the statistical office asks for a *controlled* rounding of the table, meaning that each entry has to be rounded to any of its lower/upper integer multiples of 5, so as to preserve the marginal totals in each row and in each column. As to the entries which are already multiples of 5, one typically requires that they are preserved in the final table (*zero-restrictedness* condition) so as to produce statistically unbiased rounded tables; see Figure 1(b) for an illustration.

A more detailed description of the problem is as follows. Let I be the index set of a given *table* to be protected.

The nominal values a_i ($i \in I$) in the table satisfy a given set of linear equations, say $Ma = b$ (our model can easily be extended to the case of linear inequalities). Each column of matrix M corresponds to a cell (including marginals), and each row to a link between cells. For example, in the case of k-dimensional tables with marginals the system is of the form $Ma = 0$ and gives the 1-, 2-, ..., $k-1$-way marginal projections.

As customary, for any real value z let $\lfloor z \rfloor$ and $\lceil z \rceil$ denote the lower and upper integer part of z, respectively. Given a certain *rounding base* β, we allow each table entry a_i to be rounded to

$$\tilde{a}_i \in \{\beta\lfloor a_i/\beta \rfloor, \beta\lceil a_i/\beta \rceil\}$$

This implies, in particular, that $\tilde{a}_i = a_i$ whenever a_i is an integer multiple of β. Notice that one can with no loss of generality assume that $\beta = 1$ (if this is not the case, just divide each a_i by β). Therefore, if not stated differently, we always assume $\beta = 1$ in the sequel.

Each entry a_i of the table has two associated weights, say $w_i^- \geq 0$ and $w_i^+ \geq 0$, giving a measure of the loss of information incurred if the a_i is rounded to $\lfloor a_i \rfloor$ or to $\lceil a_i \rceil$, respectively.

The (zero-restricted) *Controlled Rounding Problem* (CRP) then calls for finding a rounding $\tilde{a}_i$ of each entry a_i, such that $M\tilde{a} = b$ and the associated total rounding weight (expressed in terms of the given w_i^+ and w_i^-) is minimized.

This combinatorial optimization problem was first introduced by Bacharach (1966) in the context of replacing non-integers by integers in tabular arrays. It can be solved in polynomial time for k-dimensional tables with marginals if $k \leq 2$, but for $k \geq 3$ it belongs to the class of the strongly $\mathcal{NP}$-hard problems; see e.g., Kelly, Golden, and Assad (1990 b).

Previous works on CRP mainly concentrate on 2- and 3-dimensional tables with marginals. Cox and Ernst (1982) proved that the zero-restricted CRP associated to any 2-dimensional table with row and column marginal totals is always feasible. They also gave efficient methods for finding optimal controlled roundings. These methods are based on the transformation of CRP into a network flow problem; see Section 3.

Causey, Cox, and Ernst (1985) showed that the zero-restricted CRP on 3-dimensional tables with marginal totals is not always feasible, and gave a simple $2 \times 2 \times 2$ counter-example. Kelly, Golden, Assad, and Baker (1990) proposed a branch-and-bound procedure based on a Linear Programming for the exact solution of the problem, and addressed relaxed models.

Heuristic solution procedures have been proposed by several authors, including Causey, Cox, and Ernst (1985), and Kelly, Golden, and Assad (1990 a, 1993).

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 Statistics Netherlands, involves several research groups from both academia and national statistical offices. We participated in the project for the definition of mathematical models and solution algorithms for protecting sensitive information in tabular data. The present article describes some of the results we obtained by using the controlled rounding methodology. Results pertaining to the use of a different technique, known as the Complementary Cell Suppression, can be found in Fischetti and Salazar (1996, 1998). For both approaches, the algorithms we propose have been embedded within *τ-ARGUS*, a prototype software package for statistical disclosure control under development at Statistics Netherlands.

In this article we address mathematical models and solution algorithms for controlled rounding in the 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 hierarchical and linked tables. Moreover, we analyze the use of enlarged rounding windows to deal with the cases in which the zero-restricted version of the problem admits no feasible solution. The article is organized as follows.

In Section 2 we address the problem of finding any feasible solution of the (zero-restricted) controlled rounding problem. This is actually the main issue for many practical cases in which the objective function is not specified. We rephrase this problem as finding an integral point belonging to a certain polytope (a difficult problem in general), and address the related problem of finding an extreme point (vertex) of the same polytope.

We then consider the case in which any user-defined linear objective function giving a measure of the perturbation introduced in the rounded table, has to be minimized.

Computational results for the zero-restricted CRP on 2- and 3-dimensional tables are reported in Sections 3 and 4, respectively.

Section 5 introduces the CRP version with relaxed rounding windows, and gives computational results for 3- and 4-dimensional tables. An interesting outcome is that 4-dimensional tables often admit no zero-restricted rounding, whereas slightly enlarged rounding windows produced feasible instances in all the cases in our test bed.

2. Finding Feasible Solutions of the Zero-restricted CRP

Let $a = [a_i : I]$ be the given nominal table, viewed as a vector in $\Re^I$, and define the polytope

$$P_{CRP} = \{\tilde{a} \in \Re^I : M\tilde{a} = b, \lfloor a_i \rfloor \leq \tilde{a}_i \leq \lceil a_i \rceil \text{ for all } i \in I\}$$

containing the (possibly fractional) vectors $\tilde{a}$ which satisfy the given linear system along with the lower and upper bounds derived from rounding.

An important observation is that P_{CRP} is never empty, in that it contains the ''nominal'' vector $[a_i]$.

By construction, there is a 1-1 correspondence between the *integer* points in P_{CRP} and the feasible CRP solutions. Hence CRP essentially translates into the problem of determining an integer point inside P_{CRP}.

A somehow related problem consists in finding an extreme point (*vertex*) of P_{CRP}. If M is totally unimodular (see Nemhauser and Wolsey (1988)), as in the case of 2-dimensional tables with marginals, the two problems are in fact equivalent. Even if this is not the case, however, a vertex of P_{CRP} is likely to contain just a few fractional components (never more than the number of rows in M), hence a vertex can be viewed as a good starting point for heuristic algorithms to determine actual integer CRP solutions.

Classical linear programming theory shows that every nonempty polytope always has a vertex; see e.g., Nemhauser and Wolsey (1988). The proof of this basic result is constructive, and applies iteratively the following procedure to convert any given point of P_{CRP} into a vertex. Assume without loss of generality that the system matrix M has linear rank equal to the number of its rows, i.e., no linear equation in the system is redundant.

Given the current point $\tilde{a} \in P_{CRP}$, let

$$F = \{i \in I : \lfloor a_i \rfloor < \tilde{a}_i < \lceil a_i \rceil\}$$

contain the indexes of the fractional components of $\tilde{a}$ (those which are not equal to the prescribed lower or upper bounds). If the columns of the submatrix of M indexed by F are linearly independent, then the current point $\tilde{a}$ is a vertex of P_{CRP}, and we are done.

Otherwise, there exists a nonzero multiplier vector $[\lambda_i : i \in I]$ such that $\lambda_i = 0$ for all $i \notin F$ and $\Sigma_{i \in I} \lambda_i M_i = 0$, where M_i denotes the column of M indexed by i. Notice that such a λ can be found efficiently through well-known numerical techniques. But then for every real ϵ we have that

$$M(\tilde{a} + \epsilon\lambda) = M\tilde{a} = b$$

i.e., the point $\tilde{a} + \epsilon\lambda$ satisfies again the given linear system. In other words, λ gives a ''direction'' along which one can perturb the current point without affecting the linear system validity.

Suppose now we start with $\epsilon = 0$, and keep increasing (or decreasing) ϵ until a threshold ϵ^* is reached such that any further increase would lead to a point $\tilde{a} + \epsilon\lambda$ violating a lower or upper bound on the variables. In this situation, one can readily see that the new point $\tilde{a} + \epsilon^*\lambda$ has at least one more integer component than $\tilde{a}$, i.e., the set F associated with the new point has fewer elements. One can then replace $\tilde{a}$ by $\tilde{a} + \epsilon^*\lambda$, and repeat the procedure until the fractional support F of the current point corresponds to a set of linearly independent columns.

The above technique allows one to find efficiently a vertex of P_{CRP}. For the case of 2-dimensional tables with marginals, this vertex is guaranteed to be integral and hence corresponds to a feasible CRP solution. Moreover, in this case the method has a nice interpretation in terms of flow circulations in a certain incremental network, as discussed in the next section.

3. Zero-restricted CRP on 2-Dimensional Tables

Let us consider a 2-dimensional table $[a_{ij} : i = 0, 1, \ldots, n; j = 0, 1, \ldots, m]$ of real numbers satisfying the system $Ma = b$ $(= 0)$ given by:

$$\sum_{i=1}^{n} a_{ij} - a_{0j} = 0, \quad \text{for all } j = 0, 1, \ldots, m$$

$$\sum_{j=1}^{m} a_{ij} - a_{i0} = 0, \quad \text{for all } i = 0, 1, \ldots, n$$

where index 0 corresponds to row/column marginals.

As we have already observed, the system matrix M is totally unimodular in this case, hence every vertex of polytope P_{CRP} is integer. In this situation one can then solve CRP efficiently by applying standard linear programming techniques. Well-known efficient solution algorithms are based on a network-flow interpretation of the above linear system; see e.g., Nemhauser and Wolsey (1988) and Ahuja, Magnanti, and Orlin (1993) for the necessary background.

Consider the following (directed) network $G = (V, A)$ with $|V| = n + m + 2$ nodes. G has a *row node* r_i associated to every row i of the table, and a *column node* c_j associated to every column j of the table. The network has the following arcs:

- an arc (r_i, c_j) for every row $i \neq 0$ and every column $j \neq 0$,
- an arc (c_0, r_i) for every row $i \neq 0$,
- an arc (c_j, r_0) for every column $j \neq 0$,
- the ''grand total'' arc (r_0, c_0).

Every arc in the network then corresponds to an entry a_{ij} of a table, and has two associated lower and upper capacity bounds equal to $\lfloor a_{ij} \rfloor$ and $\lceil a_{ij} \rceil$, respectively.

By construction, there is a 1-1 correspondence between the consistent roundings of the original table and the *integer flow circulations* in the associated network. It then follows that a consistent rounding minimizing a given cost function can be found efficiently by solving a min-cost flow problem on the network.

We have implemented this idea by using the network simplex algorithm embedded in the commercial LP package CPLEX 3.0. Computational analysis has been performed on 3,000 random instances generated as in Kelly, Golden, Assad, and Baker (1990), that we solved on a PC Pentium/75 notebook.

The base number was fixed at 3, and the table entries have been generated as random integers equal to 0 (with a certain probability δ) or between 1 and 2 (with probability $1 - \delta$). The cost function was the distance between the rounded and the nominal table (the method can easily deal with any other linear objective function specified by the user).

Table 1. Average computing time, in PC Pentium/75 seconds, for finding an optimal CRP solution

$m \times n$	Percentage of zeros				
	0	25	50	75	90
100×100	1.67	1.01	0.59	0.28	0.11
200×200	9.04	6.92	4.51	2.09	0.57
300×300	25.28	18.91	12.69	5.91	1.83

Table 1 reports average computing times for several possible ''percentage-of-zeros'' densities δ (percentage of table entries whose nominal value is zero). All the instances have been solved to proven optimality within a rather short computing time.

When no cost function is given, a simpler computation can be performed to find a feasible CRP solution. This is in the spirit of the previously described procedure to detect vertices of a polytope, as it applies to the network-flow interpretation of the equation system $Ma = b$. The method needs no LP-solver, and can be implemented rather easily.

We consider the initial (feasible and fractional) flow circulation f given by $f_{ij} = a_{ij}$ for all i, j, and apply iteratively the following procedure until all the flow components become integer. We define the *incremental network* $G(f) = (V, A(f))$ associated with the current flow $[f_{ij}]$. For every arc (i,j) in G with $\lfloor a_{ij} \rfloor < f_{ij} < \lceil a_{ij} \rceil$, the incremental network has two arcs with opposite directions, namely a *forward arc* (i, j) and a *backward arc* (j, i). By construction, circuits in $G(f)$ correspond to flow re-routing, i.e., to patterns of linearly dependent columns of the system matrix M. Hence any circuit gives a ''perturbation direction'' along which one can get a new flow circulation f' with one less fractional flow component. Iterating this procedure leads to the required integer CRP solution.

The above algorithm has been implemented in C and ran on a PC Pentium/75 notebook. Table 2 reports average computing times on the same instances considered in the previous table. It can be seen that the method allows for a considerable computing time saving with respect to the use of CPLEX 3.0 network simplex algorithm.

Table 2. Average computing time, in PC Pentium/75 seconds, for finding a feasible CRP solution

$m \times n$	Percentage of zeros				
	0	25	50	75	90
100×100	0.38	0.26	0.18	0.11	0.08
200×200	3.03	1.91	1.12	0.56	0.34
300×300	10.06	6.18	3.37	1.51	0.81

4. Zero-restricted CRP on 3-Dimensional Tables

We are given a 3-dimensional table $[a_{ijk} : i = 0, 1, \ldots, n;\ j = 0, 1, \ldots, m; k = 0, 1, \ldots, p]$ of real numbers satisfying the system $Ma = b\ (= 0)$ given by:

$$\sum_{i=1}^{n} a_{ijk} - a_{0jk} = 0, \quad \text{for all } j = 0, 1, \ldots, m, \text{ and for all } k = 0, 1, \ldots, p$$

$$\sum_{j=1}^{m} a_{ijk} - a_{i0k} = 0, \quad \text{for all } i = 0, 1, \ldots, n, \text{ and for all } k = 0, 1, \ldots, p$$

$$\sum_{k=1}^{p} a_{ijk} - a_{ij0} = 0, \quad \text{for all } i = 0, 1, \ldots, n, \text{ and for all } j = 0, 1, \ldots, m$$

where, as before, zero indexes correspond to marginal entries. Notice that the above system includes both 1- and 2-way marginal projections (easier versions of the problem can deal with 1-way projections only).

Unlike the 2-dimensional case, the zero-restricted CRP on 3-dimensional tables can be infeasible; see Causey, Cox, and Ernst (1985). Moreover, Kelly, Golden, and Assad (1989) proved the $\mathcal{NP}$-hardness of the problem.

In order to determine consistent roundings with minimum distance from the nominal table, we have implemented a branch-and-bound procedure based on classical linear programming relaxation, in the vein of Kelly, Golden, Assad, and Baker (1990).

We evaluated the performance of our branch-and-bound method on random instances generated as in Kelly, Golden, Assad, and Baker (1990). We generated and solved 20,000 tables with 60 entries, according to different dimensions and density levels. In particular 1,000 tables were generated for each choice of (m, n, p) in {(15, 2, 2), (10, 3, 2), (6, 5, 2), (5, 4, 3)} and for percentage-of-zeros density in {0%, 25%, 50%, 75%, 90%}. All tables had integer entries between 0 and 2, and were rounded using base 3.

Table 3 gives the average results for the above instances. Column ''count'' gives the number of instances (out of 1,000 trials) that required branching. Column ''nodes'' gives the average number of explored nodes when branching is needed. The computing time for solving each instance in our test bed never exceeded 0.5 seconds on a PC Pentium/75.

Additional experiments have been performed on larger instances. Table 4 gives average results for tables from $4 \times 4 \times 4$ to $8 \times 8 \times 8$. Here column ''time'' gives the average

Table 3. Statistics on Kelly-Golden-Assad-Baker tables

$m \times n \times p$	Percentage of zeros	count	nodes
$15 \times 2 \times 2$	0	36	3.89
$15 \times 2 \times 2$	25	15	3.80
$15 \times 2 \times 2$	50	18	5.00
$15 \times 2 \times 2$	75	18	3.33
$15 \times 2 \times 2$	90	2	3.00
$10 \times 3 \times 2$	0	37	4.46
$10 \times 3 \times 2$	25	52	4.12
$10 \times 3 \times 2$	50	45	3.98
$10 \times 3 \times 2$	75	21	4.24
$10 \times 3 \times 2$	90	7	3.57
$6 \times 5 \times 2$	0	82	3.98
$6 \times 5 \times 2$	25	92	4.72
$6 \times 5 \times 2$	50	81	4.21
$6 \times 5 \times 2$	75	35	3.91
$6 \times 5 \times 2$	90	9	3.22
$5 \times 4 \times 3$	0	140	5.07
$5 \times 4 \times 3$	25	156	5.63
$5 \times 4 \times 3$	50	129	5.16
$5 \times 4 \times 3$	75	59	3.98
$5 \times 4 \times 3$	90	12	3.17

computing time on a PC Pentium/75 notebook (over 1,000 trials). Column ''count'' gives the number of instances requiring branching (out of the 1,000 trials). Column ''nodes'' gives the average number of nodes computed with respect to the cases requiring branching. Again, all problems were solved to optimality within short computing time.

The above figures show the effectiveness of our branch-and-bound method, which is mainly due to the fact that a vertex of the polytope P_{CRP} associated with 3-dimensional tables very likely has (almost) all integer components. Moreover, all the instances in our test bed admitted a zero-restricted controlled rounding solution.

5. Controlled Rounding with Relaxed Rounding Windows

In order to deal with the cases in which the zero-restricted CRP has no feasible solution, we propose the following model.

Let $a = [a_i : i \in I]$ be again the nominal table, satisfying a certain linear system $Ma = b$, and let $\beta = 1$ be the base number. We associate an integer variable x_i to each $i \in I$, representing a possible rounding for entry a_i. In addition, for each x_i we specify a lower and an upper bound, say lb_i and ub_i, respectively. In the classical (zero-restricted) CRP one defines $lb_i = \lfloor a_i \rfloor$ and $ub_i = \lceil a_i \rceil$. In the present model, instead, we allow some entries to have a larger rounding window $[lb_i, ub_i]$. In any case, we require $lb_i \leq a_i \leq ub_i$.

The CRP with relaxed rounding windows is now stated as the following integer LP:

$$\text{minimize} \sum_{i \in I} w_i x_i$$

Table 4. Statistics on larger 3-dimensional tables

$m \times n \times p$	Percentage of zeros	time	count	nodes
$4 \times 4 \times 4$	0	0.29	62	4.03
$4 \times 4 \times 4$	25	0.27	33	4.94
$4 \times 4 \times 4$	50	0.25	27	5.67
$4 \times 4 \times 4$	75	0.23	27	5.07
$4 \times 4 \times 4$	90	0.19	3	3.67
$6 \times 6 \times 6$	0	2.32	121	17.79
$6 \times 6 \times 6$	25	1.57	123	12.77
$6 \times 6 \times 6$	50	1.16	112	13.11
$6 \times 6 \times 6$	75	0.66	91	12.05
$6 \times 6 \times 6$	90	0.28	36	6.72
$7 \times 7 \times 7$	0	13.66	162	40.54
$7 \times 7 \times 7$	25	12.18	159	40.85
$7 \times 7 \times 7$	50	6.58	152	24.49
$7 \times 7 \times 7$	75	4.63	134	25.81
$7 \times 7 \times 7$	90	2.49	78	10.82
$8 \times 8 \times 8$	0	64.97	172	102.09
$8 \times 8 \times 8$	25	46.19	175	83.42
$8 \times 8 \times 8$	50	30.01	173	68.98
$8 \times 8 \times 8$	75	15.13	165	58.78
$8 \times 8 \times 8$	90	3.18	97	18.69

Table 5. 3-dimensional tables (10 instances for each trial)

dim	δ	time		nodes		r1	r2
$10 \times 10 \times 12$	0	47.90	(113.03)	23.9	(68)	10	0
$10 \times 10 \times 12$	25	21.75	(32.77)	9.9	(17)	10	0
$10 \times 10 \times 12$	50	13.98	(22.59)	12.2	(28)	10	0
$10 \times 10 \times 12$	75	2.54	(7.21)	8.8	(34)	10	0
$10 \times 10 \times 12$	90	0.28	(0.52)	2.6	(8)	10	0
$10 \times 10 \times 16$	0	84.13	(170.51)	18.1	(52)	10	0
$10 \times 10 \times 16$	25	59.28	(165.72)	20.6	(76)	10	0
$10 \times 10 \times 16$	50	35.14	(112.69)	19.6	(83)	10	0
$10 \times 10 \times 16$	75	5.76	(16.39)	10.8	(46)	10	0
$10 \times 10 \times 16$	90	0.44	(1.02)	3.0	(12)	10	0
$10 \times 10 \times 20$	0	131.72	(181.14)	12.4	(22)	10	0
$10 \times 10 \times 20$	25	142.51	(489.76)	26.7	(116)	10	0
$10 \times 10 \times 20$	50	55.61	(156.12)	17.1	(57)	10	0
$10 \times 10 \times 20$	75	11.52	(29.91)	12.7	(39)	10	0
$10 \times 10 \times 20$	90	1.02	(3.12)	7.8	(36)	10	0
$10 \times 12 \times 12$	0	99.15	(350.53)	29.8	(134)	10	0
$10 \times 12 \times 12$	25	54.50	(84.24)	21.9	(43)	10	0
$10 \times 12 \times 12$	50	27.40	(57.93)	18.2	(41)	10	0
$10 \times 12 \times 12$	75	3.25	(4.80)	5.8	(14)	10	0
$10 \times 12 \times 12$	90	0.38	(0.63)	2.9	(7)	10	0
$10 \times 12 \times 16$	0	260.47	(707.39)	38.0	(139)	10	0
$10 \times 12 \times 16$	25	178.70	(273.64)	33.7	(51)	10	0
$10 \times 12 \times 16$	50	59.37	(185.63)	18.6	(68)	10	0
$10 \times 12 \times 16$	75	31.38	(252.83)	64.9	(591)	10	0
$10 \times 12 \times 16$	90	0.71	(1.74)	3.8	(13)	10	0
$10 \times 12 \times 20$	0	488.20	(1,547.80)	47.0	(200)	10	0
$10 \times 12 \times 20$	25	458.52	(782.36)	64.1	(103)	10	0
$10 \times 12 \times 20$	50	143.45	(450.69)	29.4	(117)	10	0
$10 \times 12 \times 20$	75	33.66	(133.42)	33.4	(181)	10	0
$10 \times 12 \times 20$	90	0.81	(1.24)	2.3	(6)	10	0
$10 \times 14 \times 16$	0	394.83	(1,053.44)	45.1	(136)	10	0
$10 \times 14 \times 16$	25	315.57	(516.02)	39.9	(88)	10	0
$10 \times 14 \times 16$	50	144.17	(297.62)	31.5	(83)	10	0
$10 \times 14 \times 16$	75	17.86	(48.57)	13.6	(44)	10	0
$10 \times 14 \times 16$	90	1.14	(2.28)	5.1	(15)	10	0
$10 \times 14 \times 20$	0	947.80	(2,268.79)	69.0	(183)	10	0
$10 \times 14 \times 20$	25	676.15	(1,261.33)	60.6	(106)	10	0
$10 \times 14 \times 20$	50	422.20	(615.34)	63.7	(91)	10	0
$10 \times 14 \times 20$	75	38.04	(104.45)	17.8	(53)	10	0
$10 \times 14 \times 20$	90	1.42	(2.08)	3.3	(8)	10	0
$10 \times 16 \times 16$	0	800.64	(1,921.04)	62.7	(143)	10	0
$10 \times 16 \times 16$	25	463.16	(999.41)	51.5	(139)	10	0
$10 \times 16 \times 16$	50	253.62	(583.32)	42.5	(112)	10	0
$10 \times 16 \times 16$	75	143.83	(1,110.21)	121.1	(1,037)	10	0
$10 \times 16 \times 16$	90	1.51	(4.09)	5.9	(23)	10	0

Table 5. (cont)

dim	δ	time		nodes		r1	r2
10 × 16 × 18	0	1,210.74	(2,549.81)	75.9	(169)	10	0
10 × 16 × 18	25	1,129.88	(2,466.72)	95.8	(212)	10	0
10 × 16 × 18	50	358.91	(789.12)	50.1	(109)	10	0
10 × 16 × 18	75	51.99	(192.16)	23.3	(103)	10	0
10 × 16 × 18	90	1.81	(3.58)	5.5	(15)	10	0
10 × 16 × 20	0	1,602.96	(4,226.44)	79.5	(239)	10	0
10 × 16 × 20	25	1,266.72	(2,241.60)	82.0	(177)	10	0
10 × 16 × 20	50	426.43	(670.66)	46.6	(83)	10	0
10 × 16 × 20	75	91.72	(189.95)	34.3	(92)	10	0
10 × 16 × 20	90	3.42	(8.04)	9.1	(26)	10	0
10 × 18 × 18	0	1,571.32	(4,100.47)	75.1	(235)	10	0
10 × 18 × 18	25	966.47	(1,921.61)	57.3	(119)	10	0
10 × 18 × 18	50	696.01	(1,842.77)	74.3	(189)	10	0
10 × 18 × 18	75	138.73	(402.68)	50.6	(192)	10	0
10 × 18 × 18	90	2.74	(9.49)	6.7	(33)	10	0

subject to

$$Mx = b$$

$$lb_i \leq x_i \leq ub_i \quad \text{for all } i \in I$$

$$x_i \text{ integer} \quad \text{for all } i \in I$$

where one can set e.g., $w_i = 1$ if $a_i \leq (lb_i + ub_i)/2$ and $w_i = -1$ otherwise, so as to encourage rounding a cell to its nearest bound.

For the solution of the above model we have implemented a branch-and-cut scheme in the spirit of the one proposed by Padberg and Rinaldi (1991) for the solution of hard integer LP's. In our implementation, at each node of the branching tree the quality of the LP relaxation of the model is enhanced by the addition of classical Gomory cuts, as well as of the {0, 1/2}-cuts recently proposed by Caprara and Fischetti (1996).

A critical point concerns the choice of the lower/upper bounds to be imposed on each variable x_i. We conducted experiments by starting with the smallest (zero-restricted) rounding windows, and enlarging some of them if no feasible solution existed. To be more specific, we decided to always set $lb_i = \lfloor a_i \rfloor$ and $ub_i = \lceil a_i \rceil$ for the fractional entries a_i. As to the integer entries a_i, the rounding window is defined according to one of the following rules:

1. $lb_i = ub_i = a_i$ for each integer a_i (zero-restricted case);
2. $lb_i = a_i$ and $ub_i = a_i + 1$, for each integer a_i (weight w_i being set to a large positive number).

In our experiments, the second rule is only applied when the first rule does not yield any feasible controlled rounding solution, a situation arising when all the nodes of the branch-and-cut tree produced inconsistent LP relaxations of our model. Notice however that, in practice, one can use the second rule directly, as the large weights w_i assigned to the integer a_i's guarantee that an optimal solution has as few components $x_i = a_i + 1$ as possible (none if a zero-restricted solution exists).

Table 6. 4-dimensional tables (10 instances for each trial)

dim	δ	time		nodes		r1	r2
$4\times4\times4\times4$	0	0.44	(1.05)	2.8	(7)	10	0
$4\times4\times4\times4$	25	0.61	(1.47)	6.0	(16)	6	4
$4\times4\times4\times4$	50	0.45	(0.67)	1.7	(6)	1	9
$4\times4\times4\times4$	75	0.30	(0.28)	1.3	(3)	1	9
$4\times4\times4\times4$	90	0.13	(0.25)	1.4	(3)	9	1
$4\times4\times4\times6$	0	1.26	(2.30)	7.7	(15)	10	0
$4\times4\times4\times6$	25	1.90	(5.39)	22.6	(82)	8	2
$4\times4\times4\times6$	50	1.11	(3.16)	6.5	(27)	4	6
$4\times4\times4\times6$	75	0.43	(0.70)	2.9	(7)	2	8
$4\times4\times4\times6$	90	0.22	(0.32)	1.4	(4)	6	4
$4\times4\times4\times8$	0	3.33	(5.86)	13.6	(24)	10	0
$4\times4\times4\times8$	25	4.62	(9.01)	43.7	(97)	10	0
$4\times4\times4\times8$	50	2.49	(5.81)	15.6	(75)	4	6
$4\times4\times4\times8$	75	0.56	(1.00)	4.2	(14)	4	6
$4\times4\times4\times8$	90	0.23	(0.41)	2.1	(7)	8	2
$4\times4\times4\times10$	0	6.84	(10.43)	16.9	(31)	10	0
$4\times4\times4\times10$	25	7.25	(21.33)	43.3	(138)	10	0
$4\times4\times4\times10$	50	6.65	(24.39)	33.2	(135)	7	3
$4\times4\times4\times10$	75	0.91	(2.19)	1.8	(3)	1	9
$4\times4\times4\times10$	90	0.30	(0.45)	1.7	(7)	5	5
$4\times4\times6\times6$	0	7.79	(12.04)	19.2	(36)	10	0
$4\times4\times6\times6$	25	38.25	(183.34)	240.7	(1,155)	10	0
$4\times4\times6\times6$	50	9.75	(35.14)	32.5	(137)	3	7
$4\times4\times6\times6$	75	1.29	(2.38)	8.0	(25)	0	10
$4\times4\times6\times6$	90	0.28	(0.39)	1.0	(1)	4	6
$4\times4\times6\times8$	0	26.21	(40.72)	35.2	(65)	10	0
$4\times4\times6\times8$	25	118.24	(367.05)	365.7	(1,194)	10	0
$4\times4\times6\times8$	50	165.55	(613.63)	203.6	(1,100)	3	7
$4\times4\times6\times8$	75	2.55	(5.15)	9.3	(24)	0	10
$4\times4\times6\times8$	90	0.42	(0.85)	2.6	(14)	4	6
$4\times4\times6\times10$	0	152.07	(1,009.12)	182.3	(1,369)	10	0
$4\times4\times6\times10$	25	364.43	(1,485.83)	659.8	(2,746)	10	0
$4\times4\times6\times10$	50	1,186.62	(3,339.20)	2,193.0	(6,170)	5	5
$4\times4\times6\times10$	75	6.51	(12.62)	12.7	(49)	0	10
$4\times4\times6\times10$	90	2.95	(21.12)	37.3	(302)	2	8
$4\times4\times8\times8$	0	93.27	(110.25)	69.9	(86)	10	0
$4\times4\times8\times8$	25	688.74	(1,658.92)	1,017.7	(2,673)	10	0
$4\times4\times8\times8$	50	3,357.11	(9,152.79)	6,202.5	(26,268)	6	4
$4\times4\times8\times8$	75	15.47	(27.12)	30.8	(90)	0	10
$4\times4\times8\times8$	90	0.80	(2.13)	5.3	(26)	2	8
$4\times6\times6\times6$	0	90.42	(294.33)	100.8	(380)	10	0
$4\times6\times6\times6$	25	829.33	(2,630.27)	1,506.8	(4,883)	10	0
$4\times6\times6\times6$	50	1,844.17	(4,422.22)	1,597.3	(10,813)	2	8
$4\times6\times6\times6$	75	10.81	(34.60)	37.4	(129)	0	10
$4\times6\times6\times6$	90	0.73	(1.46)	3.7	(17)	0	10

Our branch-and-cut algorithm has been coded in C, by using CPLEX 3.0 as the LP-solver. Tables 5 and 6 report computational results on 3- and 4-dimensional instances, respectively (for 4-dimensional tables, system $Ma = b$ contains all 1-, 2-, and 3-marginal projections).

Random instances have been generated as in Kelly, Golden, Assad, and Baker (1990): rounding base is 3, and for each given ''percentage-of-zeros'' density $\delta \in \{0\%, 25\%, 50\%, 75\%, 90\%\}$ the internal nominal values are 0 with probability δ, and random integers in $\{1, 2\}$ with probability $1 - \delta$. We solved 10 random instances for each trial.

Tables 5 and 6 provide the following information:

dim : table dimension;

δ : percentage-of-zeros density;

time : average (maximum) computing time, in PC Pentium 75 seconds, of the overall procedure;

nodes : average (maximum) number of nodes explored by the overall procedure;

r1 : number of feasible instances with rule 1 (zero-restricted case), out of $10 - r1$ trials;

r2 : number of feasible instances with rule 2, out of 10 trials.

According to Tables 5–6, a zero-restricted solution was found for all the 3-dimensional tables in our test bed, whereas for 4-dimensional tables about 40% of the generated instances have no zero-restricted CRP solution. In any case, rule 2 was sufficient to ensure a feasible rounded solution when a zero-restricted solution did not exist.

6. References

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1993). Network Flows. Prentice Hall, Englewood Cliffs.

Bacharach, M. (1966). Matrix Rounding Problem. Management Science, 9, 732–742.

Caprara, A. and Fischetti, M. (1996). {0, 1/2}-Chvátal-Gomory Cuts. Mathematical Programming (A), 74, 221–235.

Causey, B.D., Cox, L.H., and Ernst, L.R. (1985). Applications of Transportation Theory to Statistical Problems. Journal of the American Statistical Association, 80, 903–909.

Cox, L.H. and Ernst, L.R. (1982). Controlled Rounding. INFOR, 20, 423–432.

Cox, L.H. (1987). A Constructive Procedure for Unbiased Controlled Rounding. Journal of the American Statistical Association, 82, 520–524.

Fellegi, I.P. (1972). On the Question of Statistical Confidentiality. Journal of the American Statistical Association, 67, 7–18.

Fischetti, M. and Salazar, J.J. (1996). Models and Algorithms for the Cell Suppression Problem. Proceedings of the Third International Seminar on Statistical Confidentiality, Bled, October 2–4.

Fischetti, M. and Salazar, J.J. (1998). Modeling and Solving the Cell Suppression Problem for Linearly-Constrained Tabular Data. Proceedings of the meeting Statistical Disclosure Protection '98, Lisbon, March 25–27.

Kelly, J.P., Golden, B.L., and Assad, A.A. (1990 a). Using Simulated Annealing to Solve Controlled Rounding Problems. ORSA Journal on Computing, 2, 174–185.

Kelly, J.P., Golden, B.L., and Assad, A.A. (1990 b). The Controlled Rounding Problem: Relaxations and Complexity Issues. OR Spektrum, 12, 129–138.

Kelly, J.P., Golden, B.L., and Assad, A.A. (1993). Large-Scale Controlled Rounding Using Tabu Search with Strategic Oscillation. Annals of Operations Research, 41, 69–84.

Kelly, J.P., Golden, B.L., Assad, A.A., and Baker, E.K. (1990). Controlled Rounding of Tabular Data. Operations Research, 38, 760–772.

Nemhauser, G.L. and Wolsey, L.A. (1988). Integer and Combinatorial Optimization. John Wiley and Sons, New York.

Padberg, M. and Rinaldi, G. (1991). A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. SIAM Reviews, 33, 60–100.

Willenborg, L. and De Waal, T. (1996). Statistical Disclosure Control in Practice. Lecture Notes in Statistics 111. Springer-Verlag, New York.

Received December 1997
Revised August 1998