RESEARCH REPORT SERIES
(*Statistics #2004-04)*

Tabular Statistical Disclosure Control: Optimization Techniques in Suppression and Controlled Tabular Adjustment

José H. Dulá,[1] James T. Fagan,[2] Paul B. Massell

Statistical Research Division
U.S. Bureau of the Census
Washington D.C. 20233

Report Issued: September 23, 2004

Disclaimer: This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress. The views expressed are those of the author and not necessarily those of the U.S. Census Bureau.

[1] Formerly, visiting researcher from the University of Mississippi, currently with Virginia Commonwealth University.

[2] Manufacturing and Construction Division.

Tabular Statistical Disclosure Control: Optimization Techniques in Suppression and Controlled Tabular Adjustment[1].

J.H. Dulá[2] **J.T. Fagan**[3] **P.B. Massell**[4]

U.S. Census Bureau

September 2004

Abstract. The problem of disseminating tabular data such that the amount of information provided satisfies the public need while protecting individually identifiable data is a problem in all governmental statistical agencies. The problem falls into the category of *Statistical Disclosure Control* and provides many difficult policy and technical challenges for these agencies. In order to achieve the double mission of dissemination and confidentiality protection, the agencies must balance conflicting objectives. Traditionally, agencies have relied on selective suppression of sensitive cells. Because of the difficulty of suppressing optimally and the problems that may result from publishing tables with omitted cell values, new ideas have been proposed based on selective adjustment of cell values. One such method is *Controlled Tabular Adjustment* by Cox and Dandekar [2002]. In this paper we discuss the theoretical, computational and practical issues of these two approaches to Statistical Disclosure Control.

Key Words: Statistical Disclosure Control, Cell Suppression, Controlled Tabular Adjustment.

Introduction. The mission of a governmental statistical agency is to collect, process, and disseminate statistical data. The government's on-going, long-term, effectiveness in collecting accurate data from its respondents relies critically on their trust and good will and this is attained by providing and honoring assurances to protect their identity and confidentiality. Since, at the same time, the government is interested in maximizing the amount of information disseminated by the statistical reports, there emerges a clear conflict between two issues of public interest: confidentiality and quantity of information. This creates a uniquely governmental problem, and it is the source of many difficult policy and technical challenges.

[1] This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress. The views expressed on statistical, methodological, or technical issues are those of the authors and not necessarily those of the U.S. Census Bureau.

[2] jdula@vcu.edu, Visiting Researcher from the University of Mississippi.

[3] James.T.Fagan@census.gov,Manufacturing and Construction Division, U.S.Census Bureau, Washington D.C., 20233.

[4] Paul.B.Massell@census.gov,Statistical Research Division, U.S.Census Bureau, Washington D.C., 20233.

This important trust between the government and its constituents has been codified and legislated into laws that clearly define the government's role and obligations. The charter for the modern Census Bureau includes the specific provision in Title 13 of the U.S. Code where it is specifically stated that the confidentiality of the suppliers of information for use by the Census Bureau to generate statistics will be strictly protected. The Census Bureau has operated under these laws and we can attribute to them its success in fulfilling its mission while maintaining a high degree of public trust.

Other federal statistical agencies also deal with the acquisition and dissemination of data. When these agencies obtain their data independently of the Census Bureau, they operate outside of the protections of Title 13, although in many cases they operate under laws that require some degree of protection. Recognizing the need for strong protection guarantees for information providers to these statistical agencies, Congress passed into law the *Confidentiality Information Protection and Statistical Efficiency Act* of 2002 (CIPSEA). In the bill, it is stated that

> *Pledges of confidentiality by agencies provide assurances to the public that information about individuals or organizations or provided by individuals or organizations for exclusively statistical purposes will be held in confidence...*[†]

The overarching preoccupation for protecting the confidentiality of information providers limits the amount of detail that can be released by the statistical agencies for many data products. Determining the maximum amount of detail that can be released while still protecting confidentiality provides continual policy and technical challenges for the agencies. In this paper we discuss some of the technical *Statistical Disclosure Control* solutions that have been proposed in response to the specific problem of disseminating quality statistical information while protecting the identity of its providers specifically for the case of tabular data.

Architecture of a Table. Statistical data is classified into *microdata* and *tabular* data. In this paper we are focused on issues arising in tabular data; that is, tables intended for publication that summarize microdata.

Tables can be further classified into two types: *frequency* or *count* tables and *magnitude* tables. Frequency tables present cardinal quantities of instances or occurrences of the items about the subject of the table. The entries in a magnitude table are numerical values representing an amount such as production or sales. Tables 1 and 2 are illustrations of these two types of tables:

[†] http://frwebgate2.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ347.107.pdf

	gasoline	kerosene	paraffin	diesel	*Total*
Location 1	9	11	5	7	32
Location 2	5	7	1	19	32
Location 3	26	2	32	23	83
Total	40	20	38	49	147

Table 1. Count of Refineries for Four Petroleum Products for 3 sites.

	gasoline	kerosene	paraffin	diesel	*Total*
Location 1	600	800	900	400	2700
Location 2	500	400	300	600	1800
Location 3	700	550	700	800	2750
Total	1800	1750	1900	1800	7250

Table 2. Petroleum Distillates Production (millions of gallons) for 3 sites.

Any one of the two tables above illustrates other relevant properties and characteristics of tables. These are simple 2-dimensional tables. Each value is in a position of the table called a *cell*. There are two types of cells here: interior and marginal. Marginal cells contain the sum total of a row or column. Notice that there are two types of marginals: those that contain the sum of interior cells and the one that contains the sum of the marginals themselves. This is the grand total which resides, in our example, at the lower right-hand corner of the table. A row or a column in a 2-dimensional table is called a *shaft*. When a table has two dimensions, each cell belongs to two shafts. Tables can have more than two dimensions if cells belong to more than two shafts. We can visualize a simple 3-dimensional table as being contained in a cube. In such a table there would be three types of marginals: "third level marginals" associated with totals for individual shafts, "second level marginals" where the totals of third level marginals are located and a final "first level marginal" which is the grand total.

Tables need not be simple in which case they may be hierarchical. In a hierarchical architecture, the table contains additional subtotals. This concept can be generalized to that of *linked* tables. Such tables summarize data from a common data base and are connected to each other by values at individual cells; i.e., a value at a given cell may be a marginal from another table. Often disclosure methods for magnitude tables assume that the cell values are nonnegative; however, such methods can be generalized to allow negative values.

Disclosure Control Background. Statistical Disclosure Control (or "Disclosure Avoidance" or "Disclosure Limitation") is the aspect of the generation and dissemination of statistical information concerned with the management of the risks of disclosing information about respondents in tables and microdata. The effort to limit disclosure becomes considerable when it is combined with the objective of maximizing the amount of information released. The efforts of statistical disclosure control groups produce "protected" microdata and tables that somehow provide the required protection of the identities of individual respondents.

Statistical agencies assume the existence of individuals willing to make extraordinary efforts and able to perform sophisticated analyses to extract information from protected microdata or tables beyond the intended protected levels. These individuals are referred to as "attackers" (or "intruders" or "snoopers"). The production of protected statistical products must take into account the analyses and techniques attackers might employ as part of the assessment of the protection level. Although the term used to describe these individuals might connote nefarious intent, we must assume that all consumers of the statistical products are potential attackers in this sense, and we must admit that even ourselves, at times, might be challenged to try to make inferences that will lead to revelations about protected microdata or tables beyond what appears to be available from a superficial inspection of the product.

For the case of tabular data, the process of disclosure control starts with the identification of the sensitive cells in a table. Decisions as to when a cell in a table is sensitive depend on whether the table is a frequency or magnitude table. For the case of frequency tables, a cell is defined to be sensitive when the number of respondents is less than a predefined threshold. In magnitude data, each respondent's contribution to the cell value differs. Therefore, primary disclosure rules must rely on the specific contributions of the individual respondents. Typically these considerations involve the contributors with the largest proportion of the contribution to the cell value. There are several rules. For example, the "(n, k)-rule" requires protection of a cell if n or fewer contributors account for $k\%$ or more of the total cell value.

Statistical agencies have an overriding interest in protecting their sources, especially when the data is acquired under confidentiality assurances. This is important not just to protect specific information providers from damaging disclosure or unwanted confidentiality breaches but also for the long term quality and integrity of all statistical agencies. The law recognizes this and there are specific statutes in place to provide a legal framework under which government statistical agencies must operate. Another aspect of the mission of statistical agencies is to maximize the quantity of information that is disseminated every time a product is released. These two objectives are at odds with each other and cannot be easily reconciled. Technical solutions employing mathematical models that somehow incorporate the two dissonant objectives are available. In the next section we discuss the most widely used of these techniques, cell suppression for tabular data.

Cell Suppression in Statistical Disclosure Control for Tabular Data. There is a choice of techniques for protecting the value of sensitive cells in a table. One approach is to restructure the table and combine categories. Enough collapsing of cells will eventually protect any table containing sensitive cells. Unfortunately, this protection is attained at the cost of information from entire categories and wholesale loss of detail. Frequency tables can also be perturbed using methods such as random or controlled rounding. The perturbation may be applied to the source microdata file as in "confidentiality editing" (OMB 1994).

Another protection technique that applies to both frequency and magnitude tables is the suppression of sensitive cells in the final published table. Suppression consists of the omission of sensitive cell values. It is clear that suppressing only sensitive cells may not be enough to protect the table from an attacker since simple addition and subtraction on any shaft where there is a single suppressed cell can reconstitute the original value. Therefore, additional "secondary" or "complimentary" suppressions may be necessary to attain real protection for selected sensitive cells. Even this, however, may not be enough since a determined attacker, although not able to infer the exact value of a suppressed cell, may be able to calculate an interval containing this value that is sufficiently close to violate established protection levels. Therefore, it is important to select the right suppression pattern that will not reveal from within itself any more than intended amounts of information to any conceivable attack. As we will see, this is a difficult technical problem. Suppression is the most widely practiced statistical disclosure control technique for tables and it is the topic of this section.

We will motivate and develop the technique of cell suppression for tabular data by introducing the notion of an "audit" of a protected table. Consider the following simple 2-dimensional magnitude table where the cell in position (3,3) for row "Product 3" and column "Region C" has been determined to be sensitive:

	Region			
	A	B	C	*Total*
Product 1	11	21	23	55
Product 2	15	20	35	70
Product 3	19	9	**32**	60
Total	45	50	90	185

Table 3. 2-Dimensional Unprotected Table.

The statistical agency decides to protect the sensitive cell with the following suppression pattern where the asterisks represent suppressed cells:

	A	B	C	*Total*
1	11	21	23	55
2	*	20	*	70
3	*	9	*	60
Total	45	50	90	185

Table 4. 2-Dimensional Table Protected by a Suppression Pattern.

There are a total of four suppressions in this pattern: cells (2,1), (2,3), (3,1) , (3,3). Of these, one is a "primary" suppression, the original sensitive cell, cell (3,3) . The remaining three suppressions are "secondary" or "complementary" suppressions. Note that this suppression pattern satisfies the minimum requirement that the value of the original sensitive cell cannot be discovered by simple arithmetic since all suppressions are paired in whatever row and column they appear. Note also that, as it stands, an attacker cannot distinguish between primary and secondary suppressions. This adds a measure of protection to the table.

An attacker will exploit any opportunity to extract additional information from the suppressed table. The purpose of a suppression pattern audit is to expose this additional information and test its compliance with the required protection standards for the table.

An attacker wants to know as much as possible about the original values for cells (2,1), (2,3), (3,1) , (3,3). These being unknown quantities, the first step in a mathematical approach is to assign to them variables. The table can be represented as a matrix where the four variables are y_{21}, y_{23}, y_{31}, and y_{33} as follows:

$$\begin{array}{c c} & \begin{array}{cccc} A & B & C & Tot \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ Tot \end{array} & \begin{pmatrix} 11 & 21 & 23 & 55 \\ y_{21} & 20 & y_{23} & 70 \\ y_{31} & 9 & y_{33} & 60 \\ 45 & 50 & 90 & 185 \end{pmatrix} \end{array}$$

At every row and column where a variable appears, there is an algebraic relation modeling the additivity of the table. Since variables appear in rows 2 and 3 and columns 1 and 3 there are four linear equations:

$$\begin{array}{r} \text{Row 2:} \\ \text{Row 3:} \\ \text{Col A:} \\ \text{Col B:} \end{array} \left\{ \begin{array}{lllll} y_{21} & & +y_{23} & & = 50 \\ & y_{31} & & +y_{33} & = 51 \\ y_{21} & +y_{31} & & & = 34 \\ & & y_{23} & +y_{33} & = 67 \end{array} \right.$$

Other conditions on the variables may apply in addition to having to satisfy the four linear equations. For example, it may be surmised from the nature of the table that the values for the variables are nonnegative. This, indeed, will be our assumption here. Even though this is a 4 × 4 system, it is important to note that its rank is not full. To see this note how when the linear system is expressed in matrix form after negating the last two equations:

$$\begin{pmatrix} 1 & & 1 & \\ & 1 & & 1 \\ -1 & -1 & & \\ & & -1 & -1 \end{pmatrix} \begin{pmatrix} y_{21} \\ y_{31} \\ y_{23} \\ y_{33} \end{pmatrix} = \begin{pmatrix} 50 \\ 51 \\ -34 \\ -67 \end{pmatrix}$$

A dependency among the rows is evident since their sum produces the '0 = 0' result. This means that the system actually has fewer than four independent equations and, therefore, if it has one, it has multiple solutions. As we will see, this multiplicity of solutions is related to an equivalence between suppression patterns and cycles in networks. The nonnegativity restriction on the variables does not exclude multiplicity since once a nonnegative (nontrivial) solution is at hand, it is always possible to generate alternate nonnegative solutions by the convexity of the solution set.

Any solution $y_{21} \geq 0$, $y_{23} \geq 0$, $y_{31} \geq 0$, and $y_{33} \geq 0$ to the system of linear equations above is said to generate a *consistent* table; that is, a table that maintains the additivity of all rows and columns. Naturally, the original values $y_{21} = 15$, $y_{23} = 35$, $y_{31} = 19$, and $y_{33} = 32$ satisfy the nonnegativity conditions and solve the linear system. This provides an assurance that at least one solution exists and, therefore, many other values for these four variables are possible.

The information available to an attacker consists of the system of linear equations and the fact that its solution must be nonnegative. A good place to start to try to extract useful information about the original table is to focus on certain solutions of interest. For example, if variable, y_{21}, can have many values, let us look at its largest. A formal statement of this question is to find the solution to the following problem:

$$\begin{array}{llllll} \text{maximize} & y_{21} & & & & \\ \text{such that} & & & & & \\ & y_{21} & & +y_{23} & & = 50, \\ & & y_{31} & & +y_{33} & = 51, \\ & y_{21} & +y_{31} & & & = 34, \\ & & & y_{23} & +y_{33} & = 67, \end{array}$$

$$y_{21} \geq 0, y_{23} \geq 0, y_{31} \geq 0, y_{33} \geq 0.$$

This is a well defined instance of a linear program. It is a small instance of one, and it is one that can be easily solved. The optimal objective function value to this linear program is $y^*_{21} = 34$. This indicates that, no matter what choice of values for the other three variables, in order for the table to remain consistent, the variable y_{21} will take on values that are less than or equal to 34. Since the same question can be asked of the other three variables, there are three other linear programs the solutions of which provide information about the maximum value of the variables y_{23}, y_{31}, and y_{33} and, conversely, four others that answer the symmetric questions about the minimum value attainable in order to maintain consistency in the table. The solution to the eight linear programs are summarized in the following table:

Variable	Min	Max	Actual	Mid
y_{21}	0	34	15	17
y_{23}	16	50	35	33
y_{31}	0	34	19	17
y_{33}	17	51	32	34

Table 5. Protection Intervals for Suppressed Cells.

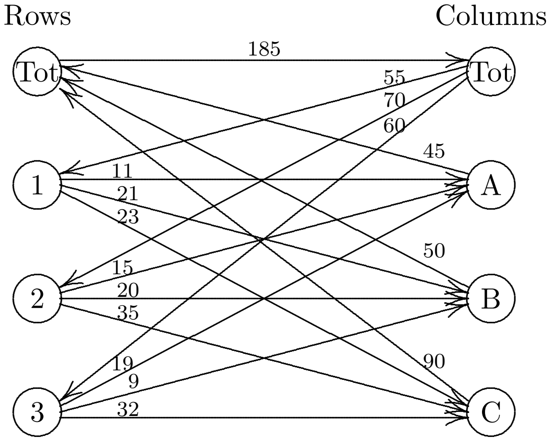
For each variable it is possible to obtain an interval in which the corresponding cell value must be located in order for the table to remain consistent. For the case of our sensitive cell in this example (cell (3,3)) the interval is $[17, 51]$. This means that an attacker cannot know anything more about this cell value given the current suppression pattern other than the fact that the original value is somewhere within this interval.

Let us focus on the protection provided by the current suppression pattern on cell (3,3). The interval $[17, 51]$ can be thought of providing a 'protection level'. This level can be quantified by taking the smaller of the two distances from the true cell value, $32 - 17$ and $51 - 32$, and calculating the corresponding percentage change it represents with respect to the original value; thus, the protection level is $46.975\% = (32 - 17)/32$ Typically, the statistical agency will define a minimum protection level p^*, to be applied symmetrically around the sensitive cell value. For obvious reasons, this minimum protection level is not publicly divulged. Any suppression pattern that generates intervals with upper and lower limits proportionately further away from the sensitive values, for all sensitive cells, than required by the official minimum protection level, p^*, is said that to "pass" the audit. Therefore, if the official minimum protection level was, say, 15%, then the current suppression pattern amply passes the audit.

A parting observation about the intervals generated by the audit based on extremal values for variables corresponding to suppressed cells is that, in this particular example, midpoints provide apparently unreliable information. For example, the original value of protected cell (3,3) is 32 whereas the midpoint of the protection interval is 34. It would appear that using the midpoint as an "estimator" may yield some information.

Suppression Patterns in 2-Dimensional Tables and Networks. All simple 2-dimensional tables have an equivalent representation using bipartite networks (see D. Gusfield (1988) and L. Cox 1995). A network is a collection of nodes and directed arcs where each arc connects a pair of nodes. In the case of bipartite graphs, the set of nodes are partitioned into two subsets and arcs connect pairs of nodes from different sides of the partition. The network representation for Table 3 is as follows:

$$\begin{array}{c|cccc} & A & B & C & Tot \\ \hline 1 & 11 & 21 & 23 & 55 \\ 2 & 15 & 20 & 35 & 70 \\ 3 & 19 & 9 & 32 & 60 \\ Tot & 45 & 50 & 90 & 185 \end{array} \quad \longrightarrow$$

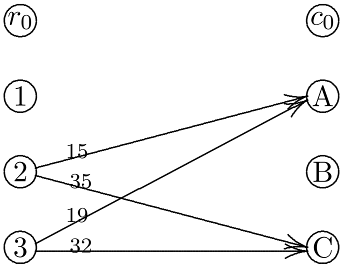


In this bipartite network, nodes on the left correspond to the rows, and nodes on the right to columns except for nodes labeled r_0 and c_0 which correspond to the marginals. The arc parameters can be treated as flows. Note that each node is "balanced" in the sense that the total flow into the node equals the total flow leaving it.

The table and the bipartite network are equivalent. There is also an equivalence between a suppression pattern in the table and a subset of the network's arcs. The arcs corresponding directly to the suppressed cells constitute the subnetwork in the bipartite graph as follows:

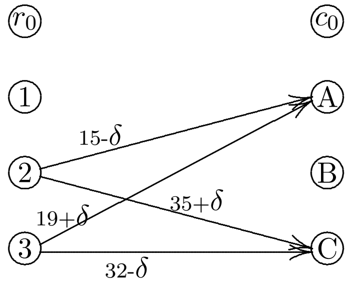
$$\begin{array}{c|cccc} & A & B & C & Tot \\ \hline 1 & 11 & 21 & 23 & 55 \\ 2 & \star & 20 & \star & 70 \\ 3 & \star & 9 & \star & 60 \\ Tot & 45 & 50 & 90 & 185 \end{array} \quad \longrightarrow$$

Suppression pattern.



Equivalent bipartite subnetwork.

Notice how we can trace a sequence of arcs in the subnetwork in the figure above as follows: 2 → C → 3 → A → 2. This sequence starts and ends at node 2 and some of the arcs are traversed against the direction of the arrow. This sequence of arcs tracing a path that starts and ends at the same node is called a *cycle.* The presence of this cycle implies a nontrivial suppression pattern and provides a way of calculating the upper and lower limits of the protection interval for any of the protected cells in the following way. Consider an increase in flow in the arc from node 2 to node C from its current value of 22. If this flow were to be increased, in order to maintain a conservation of flow into each of the nodes in the cycle there would need to be a reduction of flow into node C from node 3 which triggers a need to increase flow from 3 to A which forces a reduction in flow from 2 to A. Fortunately, this last reduction must equal the original increase from this initial node, node 2, and flow conservation is restored throughout. Graphically this sequence of adjustments caused by the original flow increase from 2 to C appears as follows:



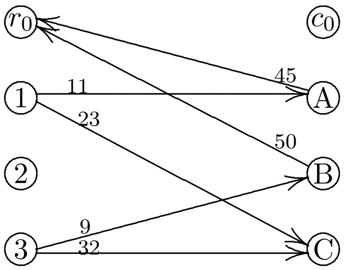
The maximum value of the flow adjustment parameter δ such that the flow remains nonnegative everywhere in the cycle is $\delta^* = 15$. Beyond this value, the flow over arc (2,A) will become negative. This means that the maximum flow possible on the arc from 2 to C is 35+δ^*=50 which agrees with the maximum value for variable y_{23} (see Table 5) obtained from solving a linear program. This same analysis can be applied to all four arcs in the subnetwork to obtain maximum and minimum flow values and these values will correspond to the solutions to the eight linear programs solved to obtain the entries in Table 5.

The equivalence between flow problems in a network and the solution to linear programs associated with suppression patterns is evidence of the general relation between suppression problems and networks. This is good news since network problems, which are a special case of linear programs†, can be solved much faster than unstructured LPs. This equivalence applies to single and 2-D tables with a hierarchical structure of any depth. The relation is lost in other types of tables. This apparently convenient relation does not mean, however, that there is an easy way to find an optimal suppression pattern.

The realization that finding an optimal suppression pattern is difficult comes from the observation that there are a combinatorial number of potential patterns in any table. Consider the simple example with which we have been working. Below are two more of the many suppression patterns and corresponding subnetworks that are possible to protect cell (3,3):

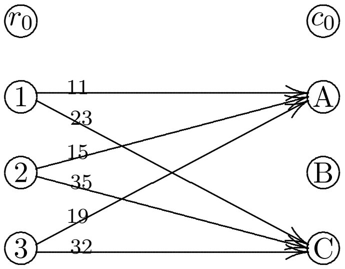
† For a reference on linear programming see Chvátal (1983).

$$\begin{array}{cc} & \begin{array}{cccc} A & B & C & Tot \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ Tot \end{array} & \begin{pmatrix} \star & 21 & \star & 55 \\ 15 & 20 & 35 & 70 \\ 19 & \star & \star & 60 \\ \star & \star & 90 & 185 \end{pmatrix} \end{array} \longrightarrow$$



Suppression Pattern 2: a unique cycle with six arcs and a marginal.

$$\begin{array}{cc} & \begin{array}{cccc} A & B & C & Tot \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ Tot \end{array} & \begin{pmatrix} \star & 21 & \star & 55 \\ \star & 20 & \star & 70 \\ \star & 9 & \star & 60 \\ 45 & 50 & 90 & 185 \end{pmatrix} \end{array} \longrightarrow$$



Suppression Pattern 3: multiple cycles.

These two new suppression patterns illustrate how all sorts of subsets of cells in a table can be used to compose a suppression pattern. They also serve to illustrate different types of patterns. In Suppression Pattern 2 the equivalent subnetwork contains more than the minimum four arcs. Suppression Pattern 3 offers an example of a more complicated situation where nodes in the subnetwork have more than two incident arcs. Notice also how it is possible to include marginal cells in a suppression pattern. Since this is at heart a network structure, such LPs can be solved at greatly accelerated speeds.

Zeroes in tables play a special role. Cells with zero values are assumed to be *structural.* In this context, structural means that no data can fall into the cell – as opposed to sampling zeroes where the cell value happens to be zero. This means that any protected table must maintain its original cells with zero values unsuppressed. Therefore, cells with zero values are never sensitive and they cannot be part of a suppression pattern.

When searching for an optimal suppression pattern in a table, one must consider all possible suppression patterns. The number of these patterns is combinatorial. Since this is a discrete problem, this search is potentially explosive. The world of combinatorial searches for optimality leads naturally to mathematical programming formulations involving binary integer variables. There are tools for solving binary integer linear programs that offer some hope for solutions for these problems.

Mathematical Programming for Optimal Cell Suppression. Searching for an optimal solution implies an objective function. Suppression causes information loss. One objective in finding an optimal suppression pattern can be to minimize a measure of information loss. This objective function must be linear if the mathematical program to be used is a Mixed Integer Linear Program (MILP).

The cell suppression problem can be modelled as a MILP formulation. This is the notation that will be used for this purpose (development and notation based on Fischetti and Salazar (2001)).

- $a = \{a_1, .., a_n\}$: the set of cell values that define the table. Note that we have switched to single subscripts. There are, therefore, n cells in this table. The cell values are the a_is.
- $\mathcal{I} = \{1, \ldots, n\}$: the index set of the cells in the table.
- $\mathcal{P} = \{i_1, \ldots, i_p\}$: the index set of the sensitive cells.
- $\mathcal{S} \subseteq \{1, \ldots, n\}$: index set of suppression pattern $\mathcal{S}$. Note, $\mathcal{P} \subset \mathcal{S}$.
- l_i and $u_i; i = 1, \ldots, n$: lower and upper limits of an interval known to any attacker as containing the value of cell i. That is, if this cell is suppressed, it is assumed that an attacker knows that the original value is in the interval $[l_i, u_i]$. Typically, statistical agencies consider the lower limit to be one half of the true cell value and the upper limit double this value although the interval may be $[0, \infty]$.

A *consistent* table with respect to the suppression pattern $\mathcal{S}$ is any set of values, $y = y_1, \ldots, y_n$, that satisfy the system:

$$\begin{aligned} My &= 0 \\ l_i \leq y_i &\leq u_i \quad \forall \quad i \in \mathcal{S} \\ y_i &= a_i \qquad \forall \quad i \notin \mathcal{S}. \end{aligned} \tag{1}$$

where the linear system $My = 0$ represents all the additivity relations of the table; one for each row and column. Other linear relations that model hierarchy can be part of this system.

Observations

1. Each of the cells in the table generates a variable in System (1).
2. The conditions on the variables $l_i \leq y_i \leq u_i$; $\forall i \in \mathcal{S}$ are necessary if the attacker knows that the original value of a suppressed cell lies within a given interval. These constraints assure that no values for the protected cells that are known to be impossible by the attacker will be considered consistent.
3. Condition $y_i = a_i$; $\forall i \notin \mathcal{S}$ means that the value of any of the variables not in the suppression pattern will take on the original cell value.

This representation of a consistent table with respect to suppression pattern $\mathcal{S}$ can be restated using binary variables. A binary variable is an integer valued variable that takes on the value of either 0 or 1. Define x_i to be a binary variable such that:

$$x_i = \begin{cases} 0 & \text{if cell } i \text{ is not suppressed} \\ 1 & \text{otherwise.} \end{cases}$$

We can use this binary variable in the following transformation:

$$\begin{cases} l_i \leq y_i \leq u_i & \forall \quad i \in \mathcal{S} \\ \quad y_i = a_i & \forall \quad i \notin \mathcal{S} \end{cases} \qquad \Longleftrightarrow \qquad a_i - L_i x_i \leq y_i \leq a_i + U_i x_i;\ \forall i \in \mathcal{I}$$

where $L_i = a_i - l_i$; $U_i = u_i - a_i$.
The system becomes:

$$\begin{gathered} My = 0 \\ a_i - L_i x_i \leq y_i \leq a_i + U_i x_i; \quad \forall i \in \mathcal{I} \end{gathered} \tag{1'}$$

Systems (1) and (1') are equivalent. The latter system, however, is an expression that can be evaluated with knowledge of values for all the variables whereas the former requires a logical inclusion/exclusion test for membership in the set $\mathcal{S}$. This makes the system with binary variables more amenable for use in a mathematical programming formulation.

In order to formulate this mathematical program we proceed to define two more parameters:

- $\mathcal{L}_{i_k}$ and $\mathcal{U}_{i_k}$: the lower and upper limits of the smallest protection interval for sensitive cell i_k. Note $\mathcal{L}_{i_k} \leq a_{i_k} \leq \mathcal{U}_{i_k}$.

We have the tools to formulate a proper MILP that will achieve the desired result. We use the model and notation of Fischetti and Salazar (2001). Note that superscripts refer to sensitive cells and a single subscript is used to identify each of the n general cells independently of the dimension of the table. The minimization is performed simultaneously for all sensitive cells.

$$
\begin{array}{llcll}
\min & \sum_i w_i x_i & & & \\
\text{s.t.}_{k=1,\ldots,p} & Mf^k & = & 0 & \\
 & f_i^k & \geq & a_i - L_i x_i & i = 1, \ldots, n; \\
 & f_i^k & \leq & a_i + U_i x_i & i = 1, \ldots, n; \\
 & f_{i_k}^k & = & a_{i_k} - \mathcal{L}_{i_k} & \\
 & & & & \\
 & Mg^k & = & 0 & \\
 & g_i^k & \geq & a_i - L_i x_i & i = 1, \ldots, n; \\
 & g_i^k & \leq & a_i + U_i x_i & i = 1, \ldots, n; \\
 & g_{i_k}^k & = & a_{i_k} + \mathcal{U}_{i_k} & \\
 & & & & \\
 & x_{i_k} & = & 1, \;\; i_k \in \mathcal{P}; &
\end{array}
$$

$$x_i \in \{0,1\} \forall i; \; f^k \geq 0, g^k \geq 0; \; k = 1, \ldots, p.$$

The mixed integer linear program above for the optimal solution of the tabular suppression problem is known as the “Kelly” formulation, for Kelly (1990). It is a general formulation that applies to any sort of table including multidimensional hierarchical tables.

Several measures of information loss have been proposed. If we associate a weight $w_i \geq 0$ with each suppressed cell, different weighting schemes provide different measures of information loss. These are some of the possibilities:

1. $w_i = 1$. In this weighting scheme the suppression of any one cell is as important as any other cell. No preference is given during optimization to the suppression of any particular cell.

2. $w_i = a_i$. Here, the weight is the actual cell value. The magnitude of the cell value guides the optimization to attempt to include low valued cells in the suppression pattern. This method might be better suited for magnitude tables. Notice how there is a beneficial tendency to change interior rather than marginal cells with this scheme since marginals in nonnegative tables have values that are at least as great as any individual cell and will tend to penalize more the objective function when they are included in a suppression pattern.

3. $w_i = \log(1 + a_i)$ or any such calculation involving logarithms. This is more likely to attenuate the effect of large fluctuation in the cell values. These types of measures are more in tune with practices in information theory.

Once the weighting scheme has been selected, the objective function equals the sum of weights for those cells in a suppression pattern being evaluated (since $x_i = 1$ for those cells in the pattern and $x_i = 0$ for those not in the pattern). Thus, the optimal value of the objective function is the minimum value of the sum of weights over all possible suppression patterns.

Since the formulation is a well defined instance of an MILP, it can be the input of an application such as CPLEX 8.0 MILP Solver (ILOG (2002)). Supposing that the solver actually returns an optimal solution to the problem, the output will be a value for each element of the vectors of variables f^k and g^k along with values for each of the binary variables x_i; $i = 1, \ldots, n$. To implement the solution it is sufficient to know the values of the binary variables. Recall that if $x_i = 0$, cell i will not be suppressed and the table will be published with its value fully disclosed. If $x_i = 1$ at optimality, then cell i will belong to the optimal suppression pattern and its value will be suppressed for publication. The values for the f^k and g^k are important in determining the results of an audit on cell k in the optimal suppression pattern. For all unsuppressed cells, these are the original values in the table. For the suppressed cells these will turn out to be the the upper (f_i^k) and lower (g_i^k) limits of the protection interval for cell i.

Access to a solver, even one as powerful as CPLEX 8.0, does not mean that an optimal, or even a feasible, solution will be found. The state-of-the-art for MILP solvers still does not guarantee finding optimal solutions in reasonable time especially for large problems and the general MILP formulation above generates massive problems even for relatively small simple tables.

Let us analyze the size of a problem resulting from the MILP formulation for tabular suppression presented above. When the matrix M contains the usual additive relations for a simple two-dimensional m by n table, the vector f^k contains $m \times n$ variables for each k, where k is the index counter for the total number of sensitive cells in the table; that number is p. Therefore, the number of variables represented by the vector f^k is $m \times n \times p$. This means that f^k represents 10^7 variables in a 100×100 table with 1000 (10%) protected cells. This represents just half of the total number of continuous variables since the same number results from counting the g^k. Consider, in addition that there is one binary variable, x_i, for each cell. The grand total in variables is therefore $2(m \times n \times p) + (m \times n) = (m \times n)(2k + 1)$.

Relaxing integer conditions in an MILP can greatly simplify solving it and may, in some cases, provide useful solutions. In the case of the optimal suppression pattern formulation, a relaxation of the binary variables is rather meaningless and still leaves behind a massive LP. All this makes the original Kelly formulation impractical for all but the smallest problems, at least for the foreseeable future.

Recent work on the optimal cell suppression problem has resulted in some advancements. Recall that simple two-dimensional tables have an underlying network structure. This structure has been successfully exploited by Fischetti and Salazar (1999). Employing the max flow/min cutset network duality theory, these researchers propose a new MILP formulation for the simple two-dimensional problem. A relaxation of the MILP can yield meaningful solutions upon the addition of several different types of constraints. The resulting formulations can have large numbers of constraints but the authors report substantial improvements in solution times on both real and synthetic problems. Some of the inherent structure of two-dimensional tables has been extended into three dimensions as network problems with side constraints in Castro (2002). Finally, the dual ideas about network cut capacities and how they constrain suppressions have been adapted to the general cell suppression problem for a new MILP in Fischetti and Salazar (2001).

The optimal cell suppression problem for tabular data is an especially intractable NP-Hard problem. The MILP formulations that are available can solve problems that, in current practices, are relatively small. Although new formulations, better MILP solvers, and faster hardware increase the scale of these problems, statistical agencies cannot count on a day in the foreseeable future when optimal approaches can be applied in production especially since the size and complication of the tables they handle are ever increasing. Production of protected tables, therefore, requires effective tools and techniques. In the next section we describe current practices for production of protected cells using suppression.

Production Practices: Sequential LP-Based Heuristics. There is a clear need for procedures to protect tables with sensitive cells that are effective and can be implemented efficiently. One of the most widely used methods for this is a heuristic for suppressing sensitive cells based on solving a sequence of LPs.

The sequential LP-based suppression heuristic originally proposed by Sande (1984) borrows on the notion of "flows" from networks to protect sensitive cells. The idea is to focus on a single sensitive cell at a time and protect it by adjusting values of other "secondary" or "complimentary" cells to maintain additivity. As with networks, the formulation uses a variable definition that measures increase and decrease in flow. For each LP that will be solved we define the following variables:

$$\begin{cases} y_i^+ & \text{Increase from current cell value, } a_i; \\ y_i^- & \text{Decrease from current cell value, } a_i. \end{cases}$$

As before, there is a set of constraints to assure the additivity of the table as well as other possible relations. These are encapsulated in the following system of linear equations:

$$M(y^+ - y^-) = 0$$

Nonnegativity is required as is the constraint that the reduction of flow can never exceed the actual cell value:

$$y^+ \geq 0; \quad y^- \geq 0; \quad y_i^- \leq a_i; \ \forall i.$$

The procedure proceeds as follows. A single specific sensitive cell, $i_{\hat{k}}$, is selected and either

$$\left\{ \begin{array}{c} y^+_{i_{\hat{k}}} = \mathcal{U}_{i_{\hat{k}}} - a_{i_{\hat{k}}} \\ y^-_{i_{\hat{k}}} = 0 \end{array} \right\} \quad \text{or} \quad \left\{ \begin{array}{c} y^+_{i_{\hat{k}}} = 0 \\ y^-_{i_{\hat{k}}} = a_{i_{\hat{k}}} - \mathcal{L}_{i_{\hat{k}}} \end{array} \right\}$$

depending on whether we wish to establish a suppression pattern to pass the audit for the upper or lower protection limit. The objective function is:

$$\min \sum_i w_i (y^+_i + y^-_i).$$

All this defines an LP the solution to which will be used to determine a suppression pattern to protect specific sensitive cell, $i_{\hat{k}}$. If the value of a flow variable is strictly positive, i.e., $y^+_i > 0$ or $y^-_i > 0$, the corresponding cell i becomes part of the pattern. Note that the pair of LPs, one where $y^+_{i_{\hat{k}}} = \mathcal{U}_{i_{\hat{k}}} - a_{i_{\hat{k}}}$ and the other where $y^-_{i_{\hat{k}}} = a_{i_{\hat{k}}} - \mathcal{L}_{i_{\hat{k}}}$, need to be solved to obtain a pattern that will protect cell $i_{\hat{k}}$.

A pattern determined by the optimal solution to the LP when, say, $y^+_{i_{\hat{k}}} = \mathcal{U}_{i_{\hat{k}}} - a_{i_{\hat{k}}}$, protects cell $i_{\hat{k}}$ because $a_{i_{\hat{k}}} = \mathcal{U}_{i_{\hat{k}}}$ is a feasible solution for the audit LP, e.g., (1). The union of the two patterns obtained when the two LPs where $y^+_{i_{\hat{k}}} = \mathcal{U}_{i_{\hat{k}}} - a_{i_{\hat{k}}}$ and $y^-_{i_{\hat{k}}} = a_{i_{\hat{k}}} - \mathcal{L}_{i_{\hat{k}}}$ therefore provide the full protection for this cell.

The effectiveness of the procedure based on the LP formulation above relies on the fact that the union of suppression patterns provide simultaneous protection for the individual cells that generate them. Therefore, the successive solution of LPs, one pair for each sensitive cell, generates a sequence of suppression patterns the union of which protects all the sensitive cells.

The performance of this approach can be improved by judicious interventions. A haphazard sequencing of the LPs and independent treatment can lead to unnecessary oversuppression. Statistical agencies practice sequencing schemes to reduce this problem. One practice is to process sensitive cells in descending order of desired protection. Protection is not usually a fixed percentage of the cell magnitude, so the order of desired protection and the order of cell magnitude may be different. Objective function cost coefficients of complementary cells selected by the solution of an LP to protect a given sensitive cell can be set to zero in the next LP. This is a way of "coercing" the optimization to recycle previously selected secondary suppressions and discourage the involvement of new ones. Suppression patterns based on solutions to a sequence of LP formulations are not guaranteed to be optimal and there may be oversuppression.

Large LPs can be solved efficiently. Moreover, the LP formulations that are also network programs are even faster to solve. This means that a sequential LP-based procedure such as the one described above can be used in production for processing large tables. This is the reason statistical agencies use this approach in their operations.

The sequential LP heuristic can be enhanced. One obvious enhancement is to take advantage of symmetry in protection intervals. When the upper and lower protection limits are equidistant from the original value, the suppression pattern obtained from setting $y^+_{i_{\hat{k}}} = \mathcal{U}_{i_{\hat{k}}} - a_{i_{\hat{k}}}$ also protects cell $i_{\hat{k}}$ from below (by "reversing" flows). The only additional precaution required in the formulation is that $y^+_i \leq a_i$, $\forall i$, in order to maintain nonnegativity. Other enhancements are based on judicious ordering of the LP sequencing and manipulation of the objective function. For example, making the sequence less haphazard may lead to less over-suppression. This is the idea behind ordering the sensitive cells by their magnitude from largest to smallest. The expectation is that larger valued sensitive cells will generate suppression patterns that can also serve for other cells especially if subsequent LPs add components to the objective function to encourage the use of cells already involved in secondary suppressions.

The intractability of optimal methods for statistical disclosure control using suppression make practical heuristics necessary. Statistical agencies, however, pay a price for having to release over-suppressed tables that deprive the public of useful information. Suppressed tables in themselves, even when optimally derived, present other drawbacks. These concerns have led to exploration for alternatives to suppression as a form of disclosure control. A recent idea is *Controlled Tabular Adjustment* (CTA) by Dandekar and Cox (2002). We will treat this method for disclosure control in the next section.

Controlled Tabular Adjustment. CTA is a recent contribution to statistical disclosure control by L. Cox and R. Dandekar and (2002). This technique addresses directly some of the problems that arise in protecting tables with sensitive cells. Its domain is precisely the same as that for cell suppression; that is, a table where a specified subset of its cells must be protected to within specific protection intervals. As we have seen, an optimal suppression pattern for such a table requires the solution of an especially difficult MILP making it impractical for production purposes for all but the smallest and simplest tables. This new technique offers a different approach which can substantially extend the scale of production to larger tables but not without a price which needs to be carefully considered.

The idea behind CTA is based on perturbation. Perturbing tables to protect cells is not new; what is innovative about CTA is that there are two different sorts of perturbations applied to a table with sensitive cells: one applies to the sensitive cells themselves and the other to the rest of the table.

CTA works on the following premise. Assuming that the specification from the statistical agency is that the true value of a sensitive cell may not be knowable to within a specific interval around its original value, replacing the value in a sensitive cell by a value greater than the interval's upper limit or less than the interval's lower limit must, therefore, satisfy the agency's protection requirement. To justify this argument consider an optimal suppression pattern. An audit on any particular sensitive cell necessarily generates an interval with a lower and upper limit that satisfies the agency's protection requirements for that cell. Since anybody can perform this audit on the protected table, knowledge of some feasible upper and lower limits on a protection interval can be considered publicly available and therefore releasing one of them in advance at the time the table is published does not compromise the security requirements imposed by the agency on this table.

The second type of perturbation applies to the nonsensitive cells and corresponds to an adjustment of their value to restore additivity in the table after having changed the values of the sensitive cells. It is hoped that this second type of perturbation is as small as possible and as much as possible applied to internal, rather than marginal, cells.

The optimal solution to an MILP provides an "optimal" CTA implementation. The formulation of this MILP uses the familiar "flow" variable framework as follows:

$$\begin{cases} y_i^+ & \text{Increase from current cell value, } a_i; \\ y_i^- & \text{Decrease from current cell value, } a_i; \end{cases}$$

for all cells $1, \ldots, n$. Also as before, we introduce a binary variable, $x_k;\ k = 1, \ldots, p$, used to determine the cell value of a sensitive cell in the final protected table:

$$x_k = \begin{cases} 1 & \text{If protected cell } k \text{ will be at its upper protection limit } \mathcal{U}_k; \\ 0 & \text{If protected cell } k \text{ will be at its lower protection limit } \mathcal{L}_k; \end{cases}$$

The variables are restricted as follows. Additivity must be maintained:

$$M(y^+ - y^-) = 0.$$

Additional restrictions on the variables depend on whether or not they correspond to sensitive cells. For variables associated with nonsensitive cells, the restrictions are simply a nonnegative lower bound and an upper bound at some specified level, u_i or l_i, depending on whether there is increase or decrease in flow at that cell. These limits are typically calculated as some percent of the original cell value. For protected cells, the flow variables are determined by the limits of the protection intervals set by the statistical agency. The values of the sensitive cells will either be greater than or equal to the upper limit of the protection interval, or less than or equal to the lower limit; the final values of the binary variables, $x_{i_k};\ k = 1, \ldots, p$, will determine this. Thus

$$\begin{array}{ll} \text{Unprotected Cells} & \text{Protected Cells} \\ \left.\begin{array}{l} 0 \le y_i^+ \le u_i \\ 0 \le y_i^- \le l_i \end{array}\right\} \forall i \notin \{i_1, \ldots, i_p\}; & \left.\begin{array}{lcc} y_{i_k}^+ & \ge & (\mathcal{U}_{i_k} - a_{i_k})x_{i_k} \\ y_{i_k}^- & \ge & (a_{i_k} - \mathcal{L}_{i_k})(1 - x_{i_k}) \end{array}\right\} \forall k \in \{1, \ldots, p\},\ x \in \{0,1\}. \end{array}$$

The objective function is as before:

$$\min \sum_i w_i(y_i^+ + y_i^-).$$

Requiring the flow variables associated with the protected cells to take on values such that the cell is either at its upper or lower protection value is restrictive. This condition may result in infeasible programs. This could happen when a table has a shaft composed entirely of either sensitive cells or zeroes (zeroes in tables cannot be perturbed). The inequality condition for these variables, however, may lead to solutions where both flow variables are strictly positive. To see this, consider the following simple two-dimensional table:

			Total
	7	5	12
	8	5	13
Total	15	10	25

The single sensitive cell is in the first column of the first row; its protection interval, $(\mathcal{L}_1,\mathcal{U}_1)$, is (6,8). The inequality condition on the flow variables for the protected cells permit the solution $x_1 = 1, y_1^+ = 1, y_1^- = 1$ with all other flow variables at zero. One way to avoid having both flow variables be positive is to modify the objective function. If the coefficients for these variables are made sufficiently large, the nontrivial presence of both pairs in a solution becomes impossible. This means, however, a loss of control on the properties of the optimal adjusted table.

If the weights, w_i, in the objective function are such that $w_i = 1, \ \forall i$, the optimal solution minimizes the absolute deviation between the original and the adjusted cells. If $w_i = a_i$ the optimal solution minimizes a weighted absolute deviation. This last measure may result in a desirable effect of favoring adjustments to interior cells rather than marginals in a table with nonnegative entries.

The MILP above to find the optimal "Controlled Tabular Adjustment" is much simpler than the one for the optimal cell suppression pattern. For one, the number of variables is many fewer than that used in the suppression formulation. The optimal CTA MILP formulation has as many continuous variables as twice the number of cells, but what is more significant, the number of binary variables is just the number of protected cells. Comparing these values with those for the optimal suppression MILP means that the CTA MILP is a much more manageable mathematical program.

Controlled Tabular Adjustment: Heuristics. Due to the combinatorial nature of CTA, a natural approach is to explore heuristics. A heuristic is a procedure that is intended to provide practical solutions to difficult problems quickly at the expense of foregoing a guarantee of optimality.

Nearly all heuristics proposed so far for CTA are based on the setting of the binary variables (which determines up/down perturbations) for the sensitive cells followed by an adjustment of the the values in the rest of the cells to produce a table that satisfies additivity as follows:

General Heuristic for Controlled Tabular Adjustment

Step 1. Set the binary variables for the sensitive cells.

Step 2. Adjust values of non-sensitive cells (including marginals) to restore additivity.

Initialization in Step 1 consists of an assignment for each sensitive cell either at (or above) its upper protection value or at (or below) its lower protection value. The total number of possible initializations for a table with p sensitive cells is 2^p; a potentially explosive number. Initializations can be performed in a number of ways. For example, Cox and Dandekar propose a random decision as to whether the value of a sensitive cell will be greater than or equal to its upper protection level or less than or equal to its lower protection level. Another idea also proposed by the original authors is to sort the sensitive cells based on their original values and proceed to instantiate above or below protection levels alternating between one and the other.

We may expect that the execution of Step 1 will destroy the table's additivity. In the original paper by Cox and Dandekar (2002) the change in the values for the rest of the cells in Step 2 is achieved by solving an LP. The variables of this LPs are positive and negative "flows" for each nonsensitive cell, including those on the margins. The variables are constrained to restore additivity and to be within standard limits, usually a percentage of the original value. The objective function is a minimization of the weighted sum of the absolute deviations. This objective function is prescribed if the formulation is to remain an LP but it may be worthwhile to explore nonlinear objective functions such as quadratics especially when the software and hardware permit it. This would allow using a measure involving square deviation as an optimization criterion. The results of this quadratic program would result in protected CTA tables with desirable second order properties.

The following small example helps explore some solutions using CTA for disclosure control. (The original table appears in Cox and Dandekar (2002)).

						Total
	200	40	50	**200**	120	610
	20	70	60	**100**	120	370
	40	90	**250**	100	30	510
	100	**150**	30	80	**150**	510
Total	360	350	390	480	420	2000

Table 6. Sample Magnitude Table from Cox and Dandekar (2002):
Sensitive cells are in bold face type.

In this table, all sensitive cells are internal. Cox and Dandekar work assuming that the protection levels required for the six sensitive cells are set at ±10%. This is also the maximum adjustment allowed (±10%) for the nonsensitive cells. The MILP formulation that provides the optimal CTA protection using the absolute deviation, "$\Sigma AbsDev$", (i.e., $w_i = 1$) minimization criterion for this table is quite easily done and results in the following protected table:

						Total
	189	36	45	**220**	120	610
	22	70	56	**90**	132	370
	37	81	**275**	90	27	510
	110	**165**	27	73	**135**	510
Total	358	352	403	473	414	2000

Table 7. Optimal Adjusted Table:$z = \min \Sigma AbsDev$ (from Cox and Dandekar (2002)).
$\Sigma AbsDev = 198$ (optimum), $\Sigma WgtAbsDev = 36120$.

A few observations are in order. A total of 23 (76.67%) of the cells have been adjusted; this includes five marginals. Since the protection values of the sensitive cells are determined by the optimal solution, their adjustment above or below the original value will not be known in advance. In this solution, four of the six sensitive cells end up being adjusted upwards.

The optimal solution proposed by Cox and Dandekar is not unique. Two other optimal solutions to the same formulation are:

Alternate Optimum 1

189	36	45	**220**	120	610
20	68	54	**90**	131	363
37	81	**275**	90	27	510
110	**165**	27	80	**135**	517
356	350	401	480	413	2000

Alternate Optimum 2

192	36	45	**220**	117	610
22	70	54	**90**	132	368
36	81	**275**	93	27	512
110	**165**	27	73	**135**	510
360	352	401	476	411	2000

Table 8. Alternate Optimal Adjusted Tables. Alt.Opt.#1: $z = \min \Sigma$ AbsDev=198 (optimum), Σ WgtAbsDev=38120.
Alt.Opt.#2: $z = \min \Sigma$ WgtAbsDev=35820 (optimum), Σ AbsDev=198 (optimum).

The three optimal solutions in Tables 7 and 8 solve the same CTA MILP formulation but display different characteristics. For example, the second alternate optimum table in Table 8 (Alt.Opt.#2) has more marginals affected by the adjustment than the other two solutions. This in spite of fact that it also minimizes the weighted absolute deviation (Σ WgtAbsDev=35820). This may surprise since minimizing the weighted absolute deviation would seem to result in fewer changes in the values of the marginals, typically the largest values in a nonnegative table. However, we must keep in mind that the constraints of the problem limiting adjustments to within 10% of the original cell value can be much more determining in an optimization problem such as this one.

Let us explore how the LP based heuristic performs on this problem. Step 1 of the heuristic requires an initialization of the protected cells at either their upper or lower protection levels. There are several ways of doing this. Two of the $2^6 = 64$ ways are to sort the protected cells from largest to smallest magnitude and execute the initialization starting with the assignment of the largest magnitude cell to either its lower or upper protection level and continue alternating between upper and lower protection level assignments. This approach was used in the two tables in Table 9. The first table was instantiated beginning with the largest magnitude cell set at its lower protection level. The second table shows what happens when this initialization sets the largest sensitive cell value to its upper protection level. Both tables are the result of solving an LP that minimizes the sum of the absolute deviation $z = \min \Sigma$ AbsDev.

Instantiation 1:
Largest Mag Adjusted at Lower Protection.

186	36	50	**220**	120	612
20	70	66	**90**	124	370
44	91	**225**	100	33	493
110	**165**	30	80	**135**	520
360	362	371	490	412	1995

Instantiation 2:
Largest Mag Adjusted at Upper Protection.

214	44	50	**180**	120	608
20	70	54	**110**	120	374
36	82	**275**	90	27	510
90	**135**	30	88	**165**	508
360	331	409	468	432	2000

Table 9. LP Heuristic Adjusted Tables. Objective Function: $z = \min \Sigma$ AbsDev.
Instantiation 1: $z = \min \Sigma$ AbsDev=214, Σ WgtAbsDev=65650.
Instantiation 2: $z = \min \Sigma$ AbsDev=222, Σ WgtAbsDev=51260.

The two adjusted tables above are quite different, although the optimization criterion in both was the same. The first initialization attains an objective function value that is less than the one in the second (214 vs 222). As we might expect, this value is greater than the minimum absolute deviation value of 198 obtained by solving the MIP to optimality. Interestingly, the weighted sum of absolute deviations, Σ WgtAbsDev, is better in the second table (51260 vs 65650). Notice too that the first table adjusts eight of the ten marginals, including the grand total. The second table leaves the grand total untouched.

When the objective function of the LP is the minimization of the weighted sum of the absolute deviations, $z = \min \Sigma$ WgtAbsDev, we get the following two tables; the first when the largest magnitude sensitive cell is instantiated at its lower protection limit and the second when it is instantiated at its upper protection limit.

Instantiation 1:
Largest Mag Adjusted at Lower Protection.

187	36	47	**220**	120	610
19	63	66	**90**	132	370
44	99	**225**	109	33	510
110	**165**	28	72	**135**	510
360	363	366	491	420	2000

Instantiation 2:
Largest Mag Adjusted at Upper Protection.

213	44	53	**180**	120	610
21	77	54	**110**	108	370
36	81	**275**	91	27	510
90	**135**	32	88	**165**	510
360	337	414	469	420	2000

Table 10. LP Heuristic Adjusted Tables. Objective Function: $z = \min \sum$ WgtAbsDev.
Instantiation 1: $z = \min \sum$ WgtAbsDev=43820, $\sum$ AbsDev=224.
Instantiation 2: $z = \min \sum$ WgtAbsDev=43820, $\sum$ AbsDev=224.

The two tables in Table 10 serve to illustrate the impact of incorporating the table's cell values as weights in the objective function. The tables in Table 9 show adjustments in 80% and 70% of the marginals, respectively. The tables in Table 10 have only 30% of their marginals adjusted. It will be interesting to explain why and how the two tables in Table 10, although differing in 73.33% of the cells, are effectively alternate optima for two different optimization problems; the one where the sum of the absolute deviations is minimized and the one where the sum of the weighted absolute deviations is minimized.

Both the MIPs and LPs formulated for the CTA solution for the Cox and Dandekar example in Table 6 found optimal solutions. This is due, in part, to the fact that the "flow" variables that determine the amount of adjustment that the final table will have are bounded above at an ample level of 10% of the non-sensitive cell values. When the constraint on the maximum adjustment for non-sensitive cells is set to 1%, the MIPs are infeasible. This implies immediately that the LPs cannot be feasible. Practical experience shows that the protection percent required for the sensitive cells should be strictly less than the maximum allowed for the nonsensitive cells. As in the small example above, nonsensitive cells having protection levels that are less than sensitive cells may result in infeasible MILPs. Testing suggests that if the two protection limits are equal, the MILP, although, still feasible, may be more time consuming to solve. When infeasibility occurs, relaxing the restrictiveness of the level of adjustment for the non-sensitive cells will eventually result in a feasible solution. Another approach may be to restrict the flow variables for internal cells only and allow marginals to have greater levels of adjustments. This, however, opens the possibility of gross distortions of the values of the marginals; a potentially undesirable effect.

The exercise of comparing CTA MIP solutions with LP-based heuristic solutions based on specific initialization schemes can begin to show the difference in the qualities of these solutions. In either case, protected cells will be adjusted to either their lower or upper protection levels; this aspect of the solution is unavoidable. It is the impact on the remaining cells that determine the quality of the solution. Heuristics such as the LP heuristic discussed here will be used when the optimal CTA formulation is impractical either because the problem is too large or the hardware or software applications to solve it are inadequate. In the next section we explore the limits posed by these factors in solving the optimal MIP formulation for CTA.

Large Scale Controlled Tabular Adjustment. The fact that the number of variables is relatively modest for the optimal CTA problem is an encouragement for attempting to solve these problems at relatively large scales and to answer the question of just how large this scale can be. An experiment was performed to answer these questions. A suite of test tables was created that would permit exploring the impact of the number of cells on the computation time. One of the test tables, 'hier13', was created by R. Dandekar. This is a 3-dimensional, hierarchical, table. The other tables, 'jims101010', 'jims151515', 'jims202020', 'jims252525', 'jims303030', and 'jims353535' were synthetically created for this experiment; all are simple three dimensional tables with 10% zero internal cells and 15% of the non-zero designated as sensitive. Note that the name of these test tables indicate the number of internal cells in each of the three dimensions; thus 'jims101010' is a 10 by 10 by 10 table with 1,000 internal cells and 1,331 total cells when the marginals are also counted.

The experiment was also used to compare hardware and different version of the CPLEX MILP solver. The hardware platforms used were:

Processor:	Speed:	Name:
UltraSpark III	750 MHz	"SRDU 11"
Pentium III	2.53 GHz	"GateWay"

All tables were processed using the interactive CPLEX (ILOG (2002)) MILP solver. Although there are options controlling the search strategy for the branch-and-cut procedure used by CPLEX for solving MILPs, the "balance optimality and feasibility" option was selected for all runs reported here since it was found to be faster than the "optimality only" option in a sample of tests. The formulation used was the optimal CTA MILP mathematical program presented above. The tables were in the form of CPLEX ".LP" files with the special "BINARY" section of the file reserved to identify the binary variables.

Access to two versions of CPLEX, 7.5 and 8.0, allowed quick tests to see how well these releases have evolved in their ability to handle mixed integer programs. A fair comparison was possible between versions 7.5 and 8.0 for the solution of the optimal CTA on table 'hier13'. The results are presented pictorially in Figure 1. These results reflect the "major enhancements" announced in the v. 8.0 release (ILOG S.A. (2003)) compared to v. 7.5 and seem to square with the advertised 40% speed increase in the new release.

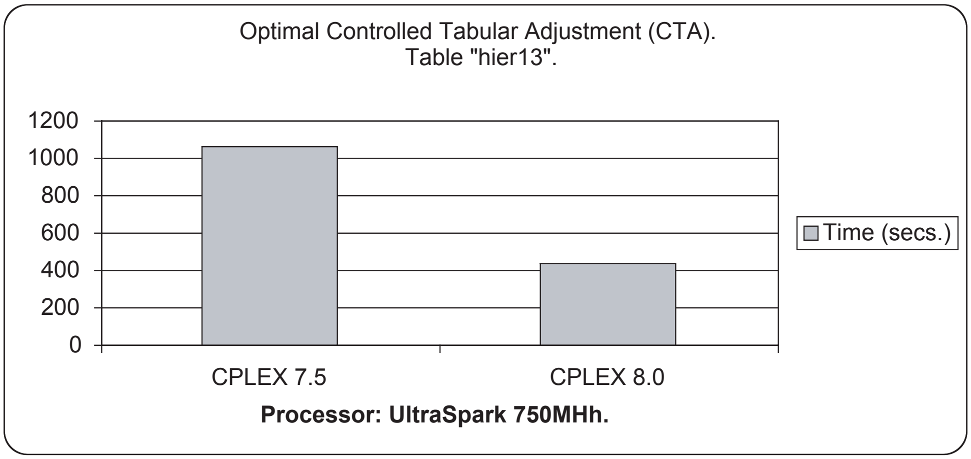


Figure 1.

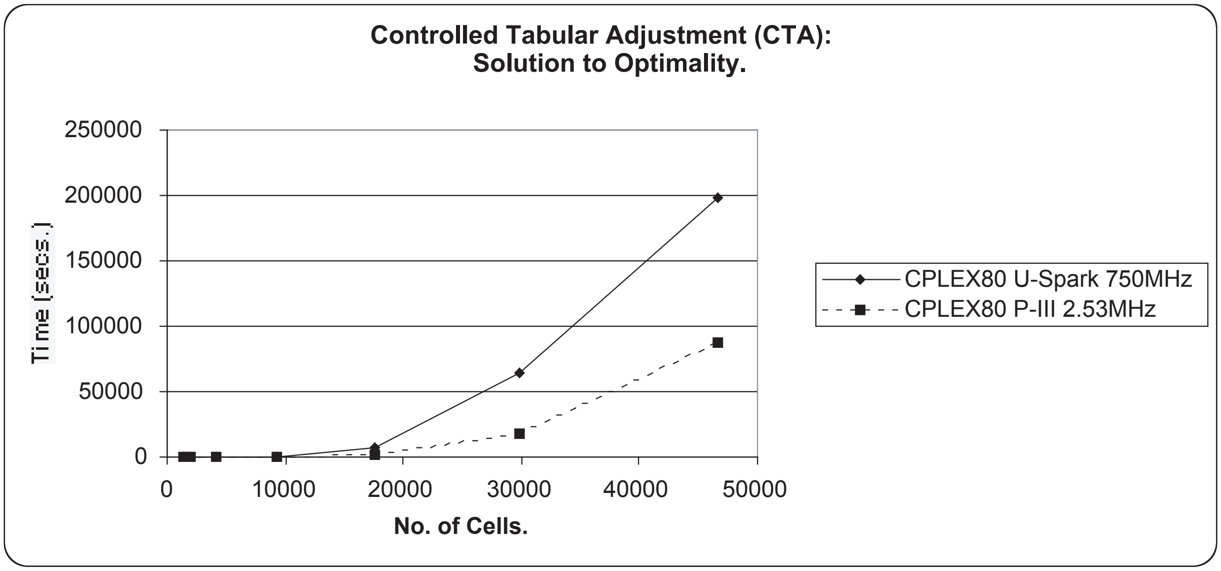


Figure 2.

The pressing question about the performance of the software and hardware is the relation between problem size and solution time. Our empirical results about this relation permit an assessment of current performance limits and extrapolation about performance limits. The synthetic data generated for these experiments yield results for a systematic performance analysis. These results are depicted in Figure 2.

The chart depicted in Figure 2 summarizes the results of experiments on the different data sets with CPLEX 8.0 and two different platforms. The first observation immediately apparent by inspection is that the Intel Pentium III based machine running at 2.53MHz performs considerably better than the "Risc" architecture running at 750MHz. The reasons for this are beyond the scope of this paper but the explanation will involve parameters such as memory and the actual CPLEX 8.0 implementations on the different machines.

Another important piece of information obtained from Figure 2 is the fact that a table with almost 50,000 cells can be solved to optimality using CTA in less than 28 hours. This may be practical for production purposes in those rare cases whenever several days have been allotted to disclosure processing of a small number of tables of this size.

A more meaningful observation about the results in Figure 2, however, is the nature of the relation between problem size and computation time. The shape of either curve is consistent with what is expected from solving MIPs; namely, an exponential growth in solution time as the problem size increases. Exponential growth is an inevitable fact in MIP due to the combinatorial nature of these problems. The next problem in the series, 'jims404040', was stopped after several days of execution. Advances in algorithms and more sophisticated software and hardware are continually extending the limits of the magnitude of MIP problems that can be solved in reasonable time although, by its nature, there will always be a sudden limit to how large these problems can be.

The final note about the MIP formulation for the optimal CTA solution proposed above refers to the objective function. Traditionally, MIPs are formulated with a linear objective function, as in the case of the formulation presented here. Until recently, this was a necessity since there were no practical computational tools to deal efficiently with nonlinear objective functions directly in MIP. This restriction is becoming less of a required practice as better algorithms and faster computers become available. Version 8.0 of CPLEX is the first of this sophisticated optimization application to allow quadratic objective functions in an MIP. This is of particular interest in optimization in statistical disclosure control since it means that we may incorporate second order information in the optimization criterion such as, say, an objective function to minimize the squared error between the original and the adjusted tables. We may anticipate that optimization results, especially with CTA, would result in adjusted protected tables possessing desirable properties that may make them better suited for certain modeling purposes. Performance of the quadratic MIP using CPLEX 8.0 needs further testing.

Metaheuristics for CTA. An optimization problem is said to be convex if the objective function is the minimization of a convex function and the feasible region is a convex set. The principal theoretical result about convex optimization problems or *convex programs* is the correspondence between local and global optima. Linear programming is the most celebrated example of a convex program. The powerful theory behind convex programs defines a clear geometry about the problems and is the main reason exact methods to solve these problems are possible; e.g., the simplex algorithm for linear programming, and the new generation of interior point methods for general convex programming. When an optimization problem is not a convex program, local optima may be difficult to define, they may abound, and any one of them may or may not be the global optimum. This is a major complication in solving non-convex programs since most optimization procedures are based on local searches directed at identifying local optima and they generally satisfy their termination criterion when this is achieved. There is no effective way to circumvent this problem and for this reason there is scant hope that there will ever be an efficient algorithm for general non-convex problems. The mixed integer formulations for cell suppression and controlled tabular adjustment, as all MIPs, involve discrete combinatorial spaces and are, therefore, nonconvex optimization problems, and herein lies the challenge.

Any effective procedure for nonconvex optimization must employ mechanisms that prevent termination simply because a local optimum has been found. Many ideas have been proposed to address this ranging from naive Monte-Carlo sampling of the domain to "intelligent" procedures that direct searches and manage memory and bookkeeping through rules for transitions to different parts of the domain to seek promising candidates for the global optimum. Examples of schemes in this last category include, among others, *simulated annealing*, *genetic algorithms*, *ant colony optimization*, and *tabu search*, and are referred to as *metaheuristics*.

Tabu search, introduced by Glover in 1989, has been adapted for use to protect tables with sensitive data using controlled tabular adjustment. The adaptation of this metaheuristic to CTA was part of a project by OptTek Systems, Inc. (2002) under contract to the Bureau of Transportation Statistics, BTS, (see also, Kelly and Russell (2003A, 2003B)). The objective of the study was to implement tabu search for CTA and explore the scale of the problems that can be solved with this technique as well as the quality of the solutions in terms of how well they compare statistically with the original values.

The input for any CTA solution approach requires a table along with the identification of its sensitive cells and corresponding protection intervals as well as ranges for each of the nonsensitive cells within which adjustment is permitted. Note that these specifications may or may not apply to marginals. A tabu search implementation for CTA also requires an objective function which, for the case of tabular data, measures the quality of an adjusted table with respect to the original data. An advantage of tabu search, and metaheuristics in general, is that they are not restricted to objective functions that must be linear or quadratic or even differentiable or analytical; any function that can be evaluated numerically can be used as an objective function in tabu search.

Any implementation of a metaheuristic to solve a particular problem requires special adaptations. This was the case of the implementation of tabu search for CTA. A basic component in tabu search is the "feasible point". For the CTA implementation, a feasible point is an adjusted table where all protected cells are instantiated at their upper or lower protection values and the nonsensitive cells are adjusted within their allowed ranges and all this, of course, while maintaining additivity. Another fundamental concept in tabu search is that of the neighborhood of a point. In the tabu implementation for CTA, a neighbor is another point obtained by changing either a sensitive or nonsensitive cell up or down: sensitive cells get changed to their protection limits, nonsensitive change by at most one unit.

In the tabu search implementation for CTA, the procedure begins by identifying a feasible point. This first procedure may involve randomization and, as a last resource, the relaxation of the adjustment limits. Once a feasible point is obtained, a second procedure begins a local search for improvements based on their definition of a neighborhood and objective function. In this procedure, cells are randomly ordered and each cell is adjusted up and down, one at a time: sensitive cells are moved to one of their protection limits and nonsensitive cells are modified by one unit. If this adjustment results in a feasible table and an improvement in the objective function, the new table is adopted as the next point in the local search in what amounts to, essentially, a simple greedy search (i.e., a search in which short term gain is emphasized without consideration of long term consequences). The process is repeated on random orderings of the cells until no improvements are found. At this point, special tabu instructions are implemented to guide the search and the iterations begin. For the case of CTA, the tabu search examines the results of a predefined number of local searches when a set of sensitive cells are anchored at their upper or lower protection levels and are not allowed to change. The decision about which combination of sensitive cells to fix as well as the number of iterations they remain in this state in the local searches is guided by performance records kept in memory and "tabu" injunctions to try to avoid previous bad performers and wasteful repetition or force possibly non-improving moves for a prespecified number of iterations.

This tabu search for CTA was implemented and tested on both simulated and real data. Compared to the solutions obtained by solving the CTA MILP to optimality using simulated tables, the tabu search implementation was able to come within 97.5% to 99.8% of the optimal objective function value with the performance improving as the size of the problem increased. Although not specified in OptTek's final report, it can be safely assumed that the results from the tabu heuristic were obtained in less time than that required for the optimal solutions. The largest problem solved optimally for this comparison was a 25×25 table. Larger simulated test problems were tested. Although no comparison with optimality was available, measures of correlation between the original and adjusted data were given as a measure of performance. When looking at one of these measures, *Correlation All* (standard linear correlation between two sets: all cells, both sensitive and nonsensitive) the authors report values of .9999 and 1.0000.

The tabu search approach was also tested on a suite of real tables provided by BTS. The performance of the metaheuristic on these problems was even more remarkable with all three measures of correlation *Correlation All*, *Correlation Sensitive*, and *Correlation Non-Sensitive* at 1.0000. A graph reporting on the time, t, to generates CTA solutions with tabu search shows there is a relation somewhere between linear and quadratic with the number of internal cells in the table, n. This relation is approximated by the function

$$t = n^{1.5}/10^6.$$

The final reading in this graph shows it takes about 1000 seconds to process a problem with 1 million internal cells.

A definite advantage of tabu search for CTA is the ability to use any objective function. This allows, for example, use of any norm, L_p, or even any ratio of norms, and not just L_1 and L_2 as when LPs or MILPs are solved. This flexibility can generate adjusted tables that are close with respect to higher order measures, e.g., $L_2, L_3, \ldots$, to the original table making them more suitable for modeling purposes. There are however, some disadvantages to this approach. Metaheuristics, by and large, do not guarantee optimality or, for that matter, any sort of bound on the proximity of a solution to optimality in any fixed amount of time. Therefore, there is no assurance that results will be acceptable; that is, the metaheuristic may or may not yield a usable adjusted table. Another problem is the reliance on probabilistic schemes at different stages of the procedure. This means that success is essentially left up to chance. There is also no guarantee of repeatability in the sense that two separate applications of the procedure may yield two different results.

Although the OptTek implementation of tabu search for CTA did not involve linear programming, there is no reason why it could not be used. There is, in fact, good reason to believe that linear programming can enhance the performance of this implementation of tabu search. The impact of LP will probably be felt in the parts of the procedure involved in finding a feasible point after some form of initialization has occurred. This is a frequently visited part of this implementation of tabu search. All indications are that using LP in the subproblems of restoring feasibility after an initialization will yield faster, better, and more conclusive results.

In spite of apprehensions about metaheuristics, there is ample empirical evidence that they work well in many difficult nonconvex problems. When properly implemented on amenable problems they can generate usable solutions in reasonable time. Preliminary results from OptTek Systems, Inc. appear to indicate that tabu search is a viable tool for protecting sensitive cells using CTA. Independent testing needs to be done to verify these claims. A separate and more difficult issue is whether or not the statistical agency believes that there are sufficient benefits to CTA for official adoption.

CTA vs Suppression. Cell suppression and controlled tabular adjustment are alternative methodologies for protecting tables with sensitive data. We have compared the two approaches in terms of technical attributes. There are, however, other considerations that should be addressed when comparing the two methodologies. The following "Pros" and "Cons" analysis addresses these considerations.

'Pros' of Cell Suppression

- Full transparency of disclosure control actions: suppressed cells speak for themselves.
- Published values are the agency's best, good faith, estimates.
- Added protection for sensitive cells since attackers cannot distinguish between primary and secondary suppressions.

'Cons' of Cell Suppression

- Optimal suppression pattern is intractable for almost all production scale applications.
- Potentially large amounts of data are removed resulting in disappointment or dissatisfaction from the user and potential negative public image.
- If sufficiently suppressed, the table is unsuitable for modeling purposes.

'Pros' of Controlled Tabular Adjustment

- Published tables are completely populated.
- Optimal adjustments are possible for relatively large tables and almost unlimited in scale for heuristics which seem to generate good quality solutions.
- Although many, and possibly all, cell values can deviate from the originals, the adjusted table, depending on the criteria used in its derivation, retain certain statistical properties that a modeler can use.
- Sensitive cells enjoy additional protection since attackers have no information on which and how data were modified.

‘Cons’ of Controlled Tabular Adjustment

- Any and all published cell values may have been modified and some may be substantially different from the original value especially if they correspond to a sensitive cell requiring a large protection interval. The inaccurate information in the table may reflect adversely on the agency’s image.

Government agencies will have a difficult decision to make when they consider either suppression or tabular adjustment to control disclosure in the tables they publish. They will have to balance many complex issues including the intended use of the information, the level of protection they afford, the technical complications of generating the protected tables, and, lastly, the impact of their decision on the agency’s image.

Conclusion. Statistical disclosure control is an inherently difficult problem because there are two opposing and contradictory objectives: to maximize the quantity and quality of information disseminated and to assure the future availability and integrity of this information by protecting the confidentiality of the sources. In this report, we have compared cell suppression with a form of perturbation, viz. controlled tabular adjustment (CTA). In terms of the final protected table, the key difference is the type of uncertainty produced in the cell values. The determination of which type of uncertainty is best depends on the policy for releasing data of the statistical office and on the perceived needs of the users of the tables. The way in which each type of protection affects the types of simple uses or statistical modeling by the users is a topic that is currently being investigated.

References

Castro, J., 2002, "Network flows heuristics for complementary cell suppression: an empirical evaluation and extensions", in *LNCS 2316, Inference Control in Statistical Databases*, J. Domingo- Ferrer (Ed), pp. 5973.

Cox, L.H., 1995, "Network models for complementary cell suppression", *J. American Statistical Association*, Vol. 75, pp. 377-385.

Cox, L. and R. Dandekar, 2002, "A disclosure limitation method for tabular data that preserves data accuracy and ease-of-use", unpublished manuscript.

Chvátal, V., 1983, *Linear Programming*, W.H. Freeman and Company, New York.

Fagan, J.T., 2001, "Cell suppression problem formulations – exact solutions and heuristics", unpublished manuscript.

Fischetti, M. and J.J. Salazar, 1999, "Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control", *Mathematical Programming*, Vol. 84, pp. 283–312.

Fischetti, M. and J.J. Salazar, 2001, "Solving the cell suppression problem in tabular data with linear constraints", *Management Science*, Vol. 47, No.7, pp. 1008-1027.

Glover, F., 1989, "Tabu search, Part 1", *ORSA Journal on Computing*, Vol. 1, pp. 190-206.

Gusfield, D., 1988, "A Graph Theoretic Approach to Statistical Data Security" *SIAM Journal on Computing*, Vol. 17, No. 3, pp. 552-571.

Kelly, J.P., 1990,"Confidentiality protection in two- and three- dimensional tables", Ph.D. Dissertation, University of Maryland, College Park, MD.

Kelly, J.P. and J.N. Russell, 2003A, "Bureau of Transportation Statistics' prototype disclosure limitation software for complex tabular data", Washington Statistical Society Seminar, January 21, 2003.

Kelly, J.P. and J.N. Russell, 2003B, "Bureau of Transportation Statistics' prototype disclosure limitation software for complex tabular data", *United Nations Statistical Commission and Economic Commission for Europe Conference of European Statisticians*, Luxembourg, April 7-9, 2003.

ILOG S.A., (2002), *ILOG CPLEX 8.0 User's Manual*, ILOG, France.

ILOG S.A. (2003) "New in ILOG CPLEX 8.0", `www.ilog.com/products/cplex/news/whatsnew.cfm`.

Massell, P.B., 2001, "Cell Suppression and audit programs used for economic magnitude data", *Research Report Series* No, RR-2001/01, Bureau of the Census Statistical Research Division, `www.census.gov/srd/papers/pdf/rr2001-01.pdf`

OptTek Systems, Inc., 2002, *Disclosure Limitation Methods for Tabular Data*, draft of final report submitted to the U.S. Department of Transportation, Bureau of Transportation Statistics, November 22, 2002.

Sande, G., 1984, "Automated Cell Suppression to Preserve Confidentiality of Business Statistics", *Statistical Journal of the United Nations ECE*, Vol. 2, pp 33-41.

Office of Management and Budget, Federal Committee on Statistical Methodology, Subcommittee on Disclosure Limitation Methodology, (1994) *Report on Statistical Disclosure Limitation Methodology*, Statistical Policy Working Paper 22, `www.fcsm.gov/working-papers/spwp22.html`.

Willenborg, L. and de Waal T., 2001, *Elements of Statistical Disclosure Control, Lecture Notes in Statistics* Vol. 155, Springer, New York.

Zayatz, L., 1992, "Using linear programming methodology for disclosure avoidance purposes", *Research Report Series* No, RR-92/02, Bureau of the Census, Statistical Research Division, `www.census.gov/srd/papers/pdf/rr92-02.pdf`.