BUREAU OF THE CENSUS
STATISTICAL RESEARCH DIVISION
RESEARCH REPORT SERIES
No. RR-92/02

USING LINEAR PROGRAMMING METHODOLOGY FOR DISCLOSURE AVOIDANCE PURPOSES

by

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

Report issued: January 16, 1992

Using Linear Programming Methodology for Disclosure Avoidance Purposes

Laura Voshell Zayatz

ABSTRACT

The Bureau of the Census is responsible for collecting information about the country's business establishments under a pledge of confidentiality and for publicly releasing this information without disclosing individual responses. The Bureau publishes the information in the form of two or three dimensional additive tables. In order to maintain the confidentiality of responses, the Bureau cannot always publish every cell value in a table. This paper describes how the Bureau uses linear programming techniques to determine which cells should be suppressed (not published) in order to publish as much information as possible while still preserving confidentiality.

KEY WORDS: Tabular Data, Linear Programming, Confidentiality

I. The Problem

The Bureau of the Census is responsible for collecting information about the country's business establishments under a pledge of confidentiality and for publicly releasing this information without disclosing individual responses. The Bureau publishes the information in the form of two or three dimensional additive tables such as those shown below. Note that all entries in the tables are **non-negative**. The values in the tables below are **fictitious**.

Two Dimensional Table

Hispanic Owned Business Enterprises
Insurance Agents, Brokers, and Service
Total Sales and Receipts in Thousands of Dollars

	All Firms	Firms with Paid Employees	Firms without Paid Employees
Total	209000	122000	87000
Mexican	96000	54000	42000
Puerto Rican	14000	7000	7000
Cuban	44000	30000	14000
Other Hispanic	55000	31000	24000

Note that in this table, values in row 1 equal the sums of values in rows 2 through 5, and values in column 1 equal the sums of values in columns 2 and 3.

Three Dimensional Table

Farms Producing Corn
Total Sales in Thousands of Dollars

Level 1 All Farms

	Delaware	New Castle County	Kent County	Sussex County
Total	30000	9200	10000	10800
Farms with 1-24 acres	4800	1400	1600	1800
Farms with 25-99 acres	6600	2000	2200	2400
Farms with 100-249 acres	8400	2600	2800	3000
Farms with 250 acres or more	10200	3200	3400	3600

Level 2 Farms with sales >= $10000

	Delaware	New Castle County	Kent County	Sussex County
Total	22200	7000	7400	7800
Farms with 1-24 acres	4200	1300	1400	1500
Farms with 25-99 acres	5100	1600	1700	1800
Farms with 100-249 acres	6000	1900	2000	2100
Farms with 250 acres or more	6900	2200	2300	2400

Level 3 Farms with sales < $10000

	Delaware	New Castle County	Kent County	Sussex County
Total	7800	2200	2600	3000
Farms with 1-24 acres	600	100	200	300
Farms with 25-99 acres	1500	400	500	600
Farms with 100-249 acres	2400	700	800	900
Farms with 250 acres or more	3300	1000	1100	1200

Note that in this table, values in row 1 equal the sums of values in rows 2 through 5, values in column 1 equal the sums of values in columns 2 through 4, and values in level 1 equal the sums of values in levels 2 and 3.

There are sometimes cell values in the tables that the Bureau cannot publish without risking a violation of the confidentiality pledge. For example, referring to the three dimensional table above, if there was only one farmer, Bob Smith, in New Castle County whose sales were greater than $10000 and who had more than 249 acres of corn, the Bureau could not publish the corresponding cell value. This is because an outsider might know that Bob Smith is the only farmer with those three characteristics and thus could see that Bob Smith had a total sales value of $2,200,000. This would be a disclosure of confidential information. The actual formula used for deciding which table cells cannot be published is confidential, however, in general, cell values that are highly dominated by one respondent are considered to possess a high risk of disclosure. The Bureau's current practice is to not publish any cell value that would enable an outsider to estimate an individual response contained in that value to within **n** percent of that response. The percent n is confidential. Any cell values that violate this criterion are called primary suppressions.

Because the tables that the Bureau publishes are additive, it is usually not enough to suppress only those cell values that violate the n percent criterion. An outsider could obtain the suppressed values through addition and subtraction. Therefore, the Bureau must suppress other cell values in the tables to ensure that an outsider cannot estimate an individual response in a primary suppression to within n percent of that response. The other values that are chosen for suppression for this reason are called complementary suppressions.

The Bureau's goal is to publish as much valuable information as possible without violating the confidentiality pledge. Thus the Bureau attempts to choose complementary suppressions in such a way that the sum of the values chosen for complementary suppression is minimized while still ensuring that the suppressions are large enough so that an individual response in a primary suppression cannot be estimated to within n percent of that response.

Consider the two dimensional additive table below.

100	12	5	250	367
12	12	5	5	34
40	200	90	300	630
5	70	50	5	130
157	294	150	560	1161

Say that there is only one business contributing to the cell value in the first row and first column. Thus, this cell is a primary suppression. We identify it as such in the table below.

P	12	5	250	367
12	12	5	5	34
40	200	90	300	630
5	70	50	5	130
157	294	150	560	1161

Say the value of n is 15 (we call this needing 15% protection). If the table above were published, an outsider could determine the exact value of the primary suppression by subtraction.

$$P = 367 - 12 - 5 - 250 = 100$$

Say we add some complementary suppressions to the table as seen below.

P	12	$\mathbf{C_{13}}$	250	367
12	12	$\mathbf{C_{23}}$	$\mathbf{C_{24}}$	34
40	200	90	300	630
$\mathbf{C_{41}}$	70	50	$\mathbf{C_{44}}$	130
57	294	150	560	1161

Using some simple algebra, an outsider could now estimate that the primary suppression value was between 95 and 105. (From Column 3 we know that $0 <= C_{13} <= 10$. Using this information and the non-negativity constraint, Row 1 implies that $85 <= P <= 105$). In other words, an outsider could estimate an individual response to within 5 percent of that response. We said that we wanted 15% protection, so we need to add more complementary suppressions, as in the table below.

P	**C**	**C**	250	367
C	**C**	**C**	**C**	34
40	200	90	300	630
C	70	50	**C**	130
157	294	150	560	1161

An outsider could now use some simple algebra or linear programming techniques to estimate that the primary suppression value was between 83 and 117. Thus, we have met our 15% protection requirement because

$$83 <= 100 - 100 * 0.15 = 85 <= P <= 100 + 100 * 0.15 = 115 <= 117.$$

In the example presented above, we said that there was only one establishment contributing to our primary suppression value. **This is not always the case.** Whenever a cell value has been designated as a primary suppression, the Bureau calculates a value **k** such that if an outsider can use algebra to at best say that

$$P - k <= P <= P + k$$

then the outsider can at best estimate any response contained in that primary suppression value to within n%. If there is only one establishment contributing to a primary suppression, then k = P * n / 100 as in the example above. When there is more than one establishment contributing to a primary suppression, the Bureau has another method of computing k. As stated before, the rule for choosing primary suppressions is confidential and the value of n is confidential. The method of calculating k is also confidential. This paper describes the technique of using linear programming to find complementary suppression patterns for a table given the cell values, the identification of certain cells as primary suppressions, and the calculated k values for those primary suppressions.

To ensure that our primary suppression in the example above was protected, we had to suppress a total cell value of 5 + 5 + 5 + 5 + 5 + 12 + 12 + 12 = 61. Note that we could have chosen a different set of complementary suppressions as shown below.

P	12	5	**C**	367
12	12	5	5	34
C	200	90	**C**	630
5	70	50	5	130
157	294	150	560	1161

This pattern provides the necessary protection, is simpler, and suppresses fewer values. But the total value of our complementary suppressions (which is what we are attempting to minimize) in this pattern is 250 + 40 + 300 = 590.

The example above shows possible complementary suppression patterns for a table with one primary suppression. Many of the Bureau's tables have several primary suppressions. If that is the case, the current practice is to choose complementary suppressions for one primary suppression at a time. We call this processing one primary suppression at a time. Each time we process a primary suppression, we suppress all cell values in the table that are chosen as complements for that primary. As one could imagine, large tables with many primary suppressions have very complicated complementary suppression patterns.

Other papers which describe this problem and/or suggest a solution to the problem are (Cox 1980), (Cox, Fagan, Greenberg, and Hemmig 1986), and (Kelly, Golden, and Assad 1990).

II. Mathematical Formulation and Explanation

Linear programming techniques can be used to find complementary suppression patterns in a table with one or more primary suppressions. **They do not yield optimal solutions.** Currently, researchers at the Bureau have not found a method for solving this problem that will always generate the best set of complementary suppressions for a table. Linear programming methods, however, do offer good solutions that ensure the n% protection requirement. The model that the Bureau uses to find complementary suppressions for a primary suppression in row r and column c in a two dimensional additive m x n table is as follows:

Decision Variables:

D_{ij1} and D_{ij2}, for all i = 1, m, j = 1, n except when (i=r **and** j=c)

Uncontrollable Variables:

$D_{rc1} \equiv$ value of k such that if an outsider can use algebra to at best say that P - k <= P <= P + k then the outsider can at best estimate any response contained in that primary suppression value to within n%.

$D_{rc2} = 0$

Constraints:

$$\sum_{i=1}^{m} (D_{ij1}-D_{ij2}) = 0 \text{ for all } j = 1, n$$

$$\sum_{j=1}^{n} (D_{ij1}-D_{ij2}) = 0 \text{ for all } i = 1, m$$

D_{ij1} <= cell value in row i, column j for all i = 1, m, j = 1, n except when (i=r **and** j=c)

D_{ij2} <= cell value in row i, column j for all i = 1, m, j = 1, n except when (i=r **and** j=c)

Objective Function:

$$\text{Minimize } \sum_{i=1}^{m} \sum_{j=1}^{n} (D_{ij1} + D_{ij2}) * \text{cost of suppressing the cell value in row i, column j}$$

where the cost of suppressing the cell value in row i, column j is calculated according to the following function:

i) 0 if the value is a primary suppression or if the value was suppressed as a complement when another primary suppression was previously processed

ii) 999999999 (a very large positive number) if the cell value is zero (the Bureau does not want to suppress any zero valued cells)

iii) the actual cell value for all other cases

Model Explanation:

Recall our example above where the cell with value 100 is a primary suppression as highlighted below.

100	12	5	250	367
12	12	5	5	34
40	200	90	300	630
5	70	50	5	130
157	294	150	560	1161

Say we suppress certain cells as complements as shown below.

P	**C**	**C**	250	367
C	**C**	**C**	**C**	34
40	200	90	300	630
C	70	50	**C**	130
157	294	150	560	1161

What does an outsider now know about the value P? An outsider could guess that the suppressed cells in the table above have the values shown in the following table.

117	**0**	**0**	250	367
0	**24**	**10**	**0**	34
40	200	90	300	630
0	70	50	**10**	130
157	294	150	560	1161

Note that although the highlighted values in this table are not the true values of the primary and complementary suppressions, the table is additive and contains only non-negative values. From this table, the outsider can see that P <= 117. P cannot be > 117, because additivity would then force one of the complements in row one to be negative.

An outsider could also guess that the suppressed cells in the table above have the values shown in the following table.

83	**24**	**10**	250	|	367
12	**0**	**0**	**10**	|	34
40	200	90	300	|	630
10	70	50	**0**	|	130
157	294	150	560	|	1161

Note again that although the highlighted values in this table are not the true values of the primary and complementary suppressions, the table is additive and contains only non-negative values. From this table, the outsider can see that P >= 83. P cannot be < 83, because additivity would then force one of the complements in row one to be larger, thereby forcing one of the complements in row two to be negative.

Thus, when the values chosen for complements above are suppressed, an outsider can use some simple algebra to estimate that the primary suppression value is between 83 and 117. He can make no better estimate of P than that. Thus, the 15% protection requirement is satisfied because

83 <= 100 - 100 * 0.15 = 85 <= P <= 100 + 100 * 0.15 = 115 <= 117.

Two valid guesses (valid in that they maintain the additivity and the non-negativity of the table) at the set of suppressed values in our example were given above, and in fact, there are many more valid ways of guessing at those values.

As stated before, the Census Bureau calculates a value **k** for each primary P such that if an outsider can use algebra to at best say that P - k <= P <= P + k, then the outsider can at best estimate any response contained in that primary suppression value to within n%. When the Bureau is attempting to find a complementary suppression pattern for a primary suppression, it makes sure that one valid guess an outsider could make at the set of suppressed values includes P = P + k and that another valid guess includes P = P - k. This ensures that an outsider can at best say that P - k <= P <= P + k.

In our model, there are two decision variables for each cell in the table, D_{ij1} and D_{ij2} for the cell in row i and column j. We will call D_{ij1} the <u>plus</u> variable and D_{ij2} the <u>minus</u> variable. Say an outsider is given a table with some suppressed values in it, and he makes a valid guess at what those values are. We define D_{ij1} and D_{ij2} for the cell in row i, column j as follows.

D_{ij1} = guessed value - true value   if guessed value >= true value
D_{ij2} = 0

or

D_{ij1} = 0   if guessed value < true value
D_{ij2} = true value - guessed value

For example, in the first valid guess described above, D_{221} = 24 - 12 = 12 and D_{132} = 5 - 0 = 5. In the second valid guess described above, D_{221} = 12 - 0 = 12 and D_{132} = 10 - 5 = 5.

Recall that our uncontrollable variables are

D_{rc1} = k
D_{rc2} = 0

These variables represent the primary suppression. In our example above, we would assign

D_{111} = 15 because 100 * 0.15 = 15
D_{112} = 0

We want to force the linear programming package to find a set of values to be suppressed as complements that will make P = P + k part of a valid guess at those values. We can think of assigning D_{rc1} = k as in effect changing the value of P in the true table to the value P + k in the outsider's table of guesses. The outsider's table of guesses must remain additive and non-negative. Because we have assigned D_{rc1} = k and we have included certain additivity constraints involving the D_{ijk}'s in our model, the linear programming package is forced to assign non-zero values to other D_{ijk}'s. If the linear programming package assigns D_{ij1} > 0, then the cell value in the in row i, column j in the true table is changed to the true cell value + D_{ij1} in the outsider's table of guesses. If the linear programming package assigns D_{ij2} > 0, then the cell value in the in row i, column j in the true table is changed to the true cell value - D_{ij2} in the outsider's table of guesses. The linear programming package assigns the D_{ijk}'s in such a way that the outsider's table of guesses is additive and non-negative.

When we run this problem through the linear programming package, the resulting values of the two decision variables representing each cell in the table fit into one of three cases.

i) $D_{ij1} = 0$ and $D_{ij2} = 0$ if the cell value in the outsider's table of guesses equals the true cell value

ii) $D_{ij1} > 0$ and $D_{ij2} = 0$ if the cell value in the outsider's table of guesses is greater than the true cell value

iii) $D_{ij1} = 0$ and $D_{ij2} > 0$ if the cell value in the outsider's table of guesses less than the true cell value

If the cell in row i, column j falls into either case ii) or case iii), the corresponding cell value has been chosen for the complementary suppression pattern.

As an example of how values in the true table may be changed to different values in an outsider's table of guesses, consider the table below where the true value of P (100) has been changed to P + k (100 + 15) as highlighted in the table below. In other words, $D_{111} = 15$. At this point, none of the other values has been changed.

115	12	5	250	367
12	12	5	5	34
40	200	90	300	630
5	70	50	5	130
157	294	150	560	1161

Note that the table is no longer additive. Some values in the table must change in order for it to be additive, and non-negativity must be maintained. The values that the linear programming package chooses to change to make the table additive will be the values suppressed as complements. One way of changing the values would be

115	12	5	**235**	367
12	12	5	5	34
25	200	90	**315**	630
5	70	50	5	130
157	294	150	560	1161

Here we have $D_{111} = 115 - 100 = 15$, $D_{142} = 250 - 235 = 15$, $D_{312} = 40 - 25 = 15$, and $D_{341} = 315 - 300 = 15$. In the table above, we have chosen to suppress as complements a total value of

$$250 + 300 + 40 = 590.$$

Another way to change the values would be

115	**2**	**0**	250	|	367
0	**22**	**10**	**2**	|	34
40	200	90	300	|	630
2	70	50	**8**	|	130
157	294	150	560	|	1161

Here we have $D_{111} = 15$, $D_{122} = 10$, $D_{132} = 5$, $D_{212} = 12$, $D_{221} = 10$, $D_{231} = 5$, $D_{242} = 3$, $D_{412} = 3$, $D_{441} = 3$. In this table, we have chosen for complementary suppression a total value of

$$12 + 12 + 12 + 5 + 5 + 5 + 5 + 5 = 61.$$

Thus, we would prefer the second suppression pattern. Both patterns above satisfy the constraints in our problem. We use the objective function to specify which pattern we prefer.

Although it may seem as if assigning $D_{rc1} = k$ will only assure that $P = P + k$ is a valid guess for P, we can use the constraints in our linear program to ensure that if $P = P + k$ is a valid guess, then $P = P - k$ is also a valid guess. This is the technique currently used by the Bureau. It is possible to use different sets of constraints and run the program twice; once to ensure that $P = P + k$ is a valid guess for P and once to ensure that $P = P - k$ is a valid guess for P. These two methods of obtaining suppression patterns can result in two different (but valid) suppression patterns. The two options will be discussed when the constraints are explained next.

The constraints for this problem can be divided into 4 groups.

i) $\sum_{i=1}^{m} (D_{ij1}-D_{ij2}) = 0$ for all j = 1, n

These constraints ensure column additivity.

ii) $\sum_{j=1}^{n} (D_{ij1}-D_{ij2}) = 0$ for all i = 1, m

These constraints ensure row additivity.

iii) D_{ij1} <= cell value in row i, column j for all i = 1, m, j = 1, n except when (i=r **and** j=c)

These are the constraints that ensure that if $P = P + k$ is a valid guess for P in the resulting table with complementary suppressions, then $P = P - k$ is also a valid guess. As stated before, when the value of the primary suppression is increased from P to $P + D_{rc1}$ in the outsider's table of guesses, other values in the outsider's table must be altered to maintain additivity. The D_{ij1} are the plus variables. They represent the values that will be increased to maintain additivity.

These constraints make sure that a cell's true value is not increased to more than twice that value in the outsider's table of guesses. Because of these constraints, we can switch the value of D_{ij1} with the value of D_{ij2} for every cell in the outsider's table of guesses and still maintain the non-negativity constraint. This means that the value of P would be changed to P - k in the outsider's table. Because we have required that D_{ij1} <= the true cell value, when we switch the D_{ij1}'s with the D_{ij2}'s, we have made sure that all D_{ij2}'s are <= the true cell value. In other words, we do not subtract more than a cell's value from that cell. In this way we ensure that if P = P + k is a valid guess for P in the resulting table with complementary suppressions, then P = P - k is also a valid guess and non-negativity in the outsider's table of guesses has been preserved.

A variation of this problem is to first ensure that P = P + k is a valid guess in the resulting table of suppressions and to then ensure that P = P - k is also a valid guess. If one desired this option, these constraints would be omitted. The program would be run once with

$D_{rc1} = k$
$D_{rc2} = 0$

to ensure that P = P + k is a valid guess in the resulting table of suppressions. It would then be run again with

$D_{rc1} = 0$
$D_{rc2} = k$

to ensure that P = P - k is also a valid guess. All cells with either D_{ij1} or D_{ij2} > 0 in either run would be suppressed.

iv) D_{ij2} <= cell value in row i, column j for all i = 1, m, j = 1, n except when (i=r **and** j=c)

These constraints enforce non-negativity. They ensure that cell values are not decreased by more than their original value in an outsider's table of guesses.

As stated before, this approach to the problem of finding complementary suppression patterns does not always yield the optimal solution. One reason for this is that the objective function that the linear programming package minimizes is

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (D_{ij1} + D_{ij2}) * \text{cost of suppressing the cell value in row } i\text{, column } j$$

What we would like minimized is

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (R_{ij1} + R_{ij2}) * \text{cost of suppressing the cell value in row } i\text{, column } j$$

where $R_{ijk} = 1$ if $D_{ijk} > 0$ and $R_{ijk} = 0$ if $D_{ijk} = 0$ (k=1,2).

Some examples of the problems that occur because of this difference in objective functions and our attempts to correct these problems are described in the Recommendations section.

Linear programming methodology can also be used to find complementary suppressions in three dimensional additive tables. The model that the Bureau uses to find complementary suppressions for a primary suppression in row r, column c, and level l in a three dimensional additive m x n x p table is as follows.

Decision Variables:

D_{ijk1} and D_{ijk2}, for all i = 1, m, j = 1, n, k = 1, p except when (i=r **and** j=c **and** k=l)

Uncontrollable Variables:

D_{rcl1} = value of k such that if an outsider can use algebra to at best say that P - k <= P <= P + k then the outsider can at best estimate any response contained in that primary suppression value to within n%.
D_{rcl2} = 0

Constraints:

$$\sum_{i=1}^{m} (D_{ijk1}-D_{ijk2}) = 0 \text{ for all } j = 1, n, k = 1, p$$

$$\sum_{j=1}^{n} (D_{ijk1}-D_{ijk2}) = 0 \text{ for all } i = 1, m, k = 1, p$$

$$\sum_{k=1}^{p} (D_{ijk1}-D_{ijk2}) = 0 \text{ for all } i = 1, m, j = 1, n$$

D_{ijk1} <= cell value in row i, column j, level k for all i = 1, m, j = 1, n, k = 1, p except when (i=r **and** j=c **and** k=l)

D_{ijk2} <= cell value in row i, column j, level k for all i = 1, m, j = 1, n, k = 1, p except when (i=r **and** j=c **and** k=l)

Objective Function:

$$\text{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} (D_{ijk1} + D_{ijk2}) * \text{cost of suppressing the cell value in row } i\text{, column } j\text{, level } k$$

where the cost of suppressing the cell value in row i, column j, level k is calculated according to the following function.

i) 0 if the value is a primary suppression or if the value was suppressed as a complement when another primary suppression was previously processed

ii) 999999999 (a very large positive number) if the cell value is zero (the Bureau does not want to suppress any zero valued cells)

iii) the actual cell value for all other cases

Model Description:

This model is simply an extension of the one for two dimensional tables. The same explanation applies.

III. An Example

Because of confidentiality reasons, we are not allowed to use real Census Bureau data to give an example of the techniques described above. We are therefore forced to use fictitious data in our example. We will use the two dimensional table described above to show how linear programming can be used to find a complementary suppression pattern that protects the response in the primary suppression in row 1, column 1.

See the LINDO program and solution in the Appendix. As stated earlier, the objective function minimized by the linear programming package really has no meaning for us. We are interested in which decision variables have been assigned non-zero values, in other words, which variables are in the basis and are non-zero. The corresponding table cells of those variables will be suppressed.

Non-Zero Variables	Corresponding Table Cells To Be Suppressed
D_{122}	Row 1, Column 2
D_{132}	Row 1, Column 3
D_{212}	Row 2, Column 1
D_{221}	Row 2, Column 2
D_{231}	Row 2, Column 3
D_{242}	Row 2, Column 4
D_{412}	Row 4, Column 1
D_{441}	Row 4, Column 4
D_{111}	Row 1, Column 1 (the Primary Suppression)

Note that this suppression pattern is the same pattern that was shown earlier. The total value of the cells that we suppress as complements in this table is 5 + 5 + 5 + 5 + 5 + 12 + 12 + 12 = 61. The resulting suppression pattern and corresponding table of guesses appear below.

P	**C**	**C**	250	367
C	**C**	**C**	**C**	34
40	200	90	300	630
C	70	50	**C**	130
157	294	150	560	1161

115	**2**	**0**	250	367
0	**22**	**10**	**2**	34
40	200	90	300	630
2	70	50	**8**	130
157	294	150	560	1161

IV. Interpretation of Sensitivity Analysis and Dual Variables

The sensitivity analysis of the cost coefficients in the objective function tells us the amount by which a cost coefficient can change without altering the optimal solution given that everything else remains constant. These values are not very significant for this application of linear programming. This is because the linear programming package is working with the D_{ijk} variables, and we are interested in the R_{ijk} variables. The R_{ijk} variables tell us which D_{ijk} variables are in the basis and are non-zero. The non-zero variables that are in the basis correspond to the cell values in the table that should be suppressed.

If one of the cost coefficients was changed by an amount that put it outside of the allowable range suggested in the sensitivity analysis, then the D_{ijk} values given by the linear programming package would change. The R_{ijk} values, on the other hand, might change, yielding a different suppression pattern. However, they might remain the same, yielding the same suppression pattern.

For our example, the sensitivity analysis shows us that if we increase the cost coefficient of the variable D_{132} by an amount >= 14, our optimal solution will change. Let's say we change the cost coefficient of the variable D_{132} from 5 to 20. Then the linear programming package would choose the following suppression pattern and corresponding table of guesses for the problem:

P	**C**	**C**	250	367
C	**C**	**C**	**C**	34
40	200	90	300	630
C	70	50	**C**	130
157	294	150	560	1161

115	**0**	**2**	250	367
0	**24**	**8**	**2**	34
40	200	90	300	630
2	70	50	**8**	130
157	294	150	560	1161

Note that the suppression pattern is the same pattern that was given when the cost coefficient of D_{132} was 5. The D_{ijk} variables have changed in the solution, but the R_{ijk} variables have not.

The sensitivity analysis of the right-hand-side values has much more meaning for us. It tells us the amount by which a right-hand-side value can change without changing our basis. If a right-hand-side value was changed by an amount that put it outside of the allowable range suggested in the sensitivity analysis, then some D_{ijk}'s which were originally > 0 would now = 0, and some D_{ijk}'s which were originally = 0 would now be > 0. Thus, some R_{ijk}'s which were originally = 1 would now = 0, and some R_{ijk}'s which were originally = 0 would now be = 1. Therefore, changing a right-hand-side value by an amount that puts it outside of the allowable range results in a different basis and a different complementary suppression pattern.

For example, in our sensitivity analysis, we see that the allowable increase for the right-hand-side of constraint number 21 is 3. Say we increase the right-hand-side for that constraint by 4. In other words, we change constraint 21 from $D_{212} <= 12$ to $D_{212} <= 16$. Then the linear programming package would choose the following suppression pattern and corresponding table of guesses for the problem.

P	**C**	**C**	250	367
C	**C**	**C**	5	34
40	200	90	300	630
5	70	50	5	130
157	294	150	560	1161

115	**0**	**2**	250	367
-3	**24**	**8**	5	34
40	200	90	300	630
5	70	50	5	130
157	294	150	560	1161

Note that this suppression pattern is indeed different from the one we obtained previously. Also note that the table of guesses is no longer non-negative. Maintaining non-negativity in this table was, in fact, the reason for having the constraint $D_{212} <= 12$.

The dual variables represent the value of an additional unit of a resource. For this problem, the dual variables for the additivity constraints represent the value in terms of the objective function of allowing the sum of the internal row (or column) values in the table of guesses to be one unit greater than the row (or column) marginals in that table. This is very abstract, and really means nothing to us. The additivity constraints absolutely cannot be changed.

In this problem, our only true resources are the D_{ijk}'s. Constraining the D_{ijk}'s has the effect of constraining the amount by which a value in the outsider's table of guesses can differ from the true value. We have constrained the D_{ijk}'s to be <= to their corresponding data values. A dual variable corresponding to the constraint D_{ijk} <= some value **v**, represents the value in terms of the objective function of changing the constraint to $D_{ijk} <= v + 1$.

For example, the dual variable for the constraint $D_{212} <= 12$ is equal to 3. If we change this constraint to $D_{212} <= 13$, the linear programming package would choose the following suppression pattern and corresponding table of guesses for the problem.

P	**C**	**C**	250	367
C	**C**	**C**	**C**	34
40	200	90	300	630
C	70	50	**C**	130
157	294	150	560	1161

115	**2**	**0**	250	367
-1	**22**	**10**	**3**	34
40	200	90	300	630
3	70	50	**7**	130
157	294	150	560	1161

The value of the objective function for this solution is 476 which is equal to the value of our original objective function (479) minus the value of the dual variable for the changed constraint (3). Therefore, by increasing our resource D_{212} by 1 unit, we have lowered the value of our objective function by 3. Note that by changing the constraint, we lose the non-negativity of the table.

V. Recommendations for Improving Solutions

As stated earlier, the linear programming technique for applying complementary suppressions to a table as described in this paper gives good results that achieve the n% protection requirement, but the results are not optimal. There are three ways of improving the results that we recommend.

One way of improving the results is to sort the primary suppression values in the table from largest to smallest and process the largest one first, the second largest second, and so on. Processing the primary suppressions in this manner tends to decrease both the number and the total value of complementary suppressions. The reason for this is that often the complementary suppressions that are chosen to protect the larger primary suppressions also provide adequate protection for the smaller primary suppressions. Thus, when the smaller primary suppressions are processed, no new complementary suppressions are needed.

On the other hand, if the small primary suppressions are processed first, very often small values will be chosen as complements.

Then, when the larger primaries are processed, many new larger complementary suppressions are needed. The result can be a table with many unnecessary small complementary suppressions. For example, see the table below where the two primary suppressions are highlighted.

200	**1000**	500	1700
50	40	400	490
80	90	500	670
200	200	600	1000
530	1330	2000	3860

If we process the primary suppression with value 200 first and the other primary second, the resulting table is

200	**1000**	500	1700
C	**C**	400	490
80	90	500	670
200	200	600	1000
530	1330	2000	3860

after processing the primary with value 200 and

200	**1000**	500	1700
C	**C**	400	490
80	90	500	670
C	**C**	600	1000
530	1330	2000	3860

after processing the primary with value 1000. If, instead, we process the primary suppression with value 1000 first and the other primary second, the resulting table is

200	**1000**	500	1700
50	40	400	490
80	90	500	670
C	**C**	600	1000
530	1330	2000	3860

after processing the primary with value 1000, and it remains

200	**1000**	500	1700
50	40	400	490
80	90	500	670
C	**C**	600	1000
530	1330	2000	3860

after processing the primary with value 200. By processing the largest primary suppression first, we eliminate the superfluous small complementary suppressions.

A second method of improving our solution is to process each primary suppression in two steps requiring two runs through the linear programming package, one with the cost function as defined in the Mathematical Formulation section and a second with the adjusted cost function described below.

i) 0 if the value is a primary suppression or if the value was suppressed as a complement when another primary suppression was previously processed

ii) 99999999 (a large positive number) if the cell value was not chosen for suppression in the first run through the linear programming package and case i) does not apply

iii) (1/cell value) if the cell value was chosen as a complementary suppression and case i) does not apply

The second run of the problem through the linear programming package often eliminates some superfluous small complementary suppressions.

Recall that the objective function that the linear programming package minimizes is

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (D_{ij1} + D_{ij2}) * \text{cost of suppressing the cell value in row } i\text{, column } j$$

This is the sum over all values of the products of the **amount** that a value is altered in the outsider's table of guesses and the cost of suppressing that value. We would like to minimize

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (R_{ij1} + R_{ij2}) * \text{cost of suppressing the cell value in row } i\text{, column } j$$

where $R_{ijk} = 1$ if $D_{ijk} > 0$ and $R_{ijk} = 0$ if $D_{ijk} = 0$ $(k=1,2)$. This is the sum of the costs of all altered values. We are not concerned about the amount by which a value is altered, only whether or not it is altered. This difference can lead to the problem shown in the example below where the primary suppression is highlighted. Say that, as before, the k value calculated for the primary suppression is 15.

100	5	20	125
5	5	50	60
20	70	20	110
125	80	90	295

When we run this problem through the linear programming package, we will get the following altered table and complementary suppression pattern.

115	**0**	**10**	125
0	**10**	50	60
10	70	**30**	110
125	80	90	295

P	**C**	**C**	125
C	**C**	50	60
C	70	**C**	110
125	80	90	295

The value of the linear programming package's optimal objective function for this example is

5 * 5 + 5 * 5 + 5 * 5 + 10 * 20 + 10 * 20 + 10 * 20 = 675.

Our objective function for this chosen suppression pattern would be

5 + 5 + 5 + 20 + 20 + 20 = 75.

Note that the three cells with value 5 do not need to be suppressed. Another valid suppression pattern is

115	5	**5**	125
5	5	50	60
5	70	**35**	110
125	80	90	295

P	5	**C**	125
5	5	50	60
C	70	**C**	110
125	80	90	295

The value of the linear programming package's objective function for this solution would be

15 * 20 + 15 * 20 + 15 * 20 = 900.

Our objective function for this solution is

20 + 20 + 20 = 60.

Thus, we prefer the second solution. The first run of this problem through the linear programming package would give us the first solution. When we run the problem through the linear programming package a second time with the costs of the cells with value 5 being changed to 1/5 and the costs of the cells with value 20 being changed to 1/20, the linear programming package will calculate a cost function of

5 * 1/5 + 5 * 1/5 + 5 * 1/5 + 10 * 1/20 + 10 * 1/20 + 10 * 1/20 =

4.5

for the first solution and

15 * 1/20 + 15 * 1/20 + 15 * 1/20 = 2.25

for the second solution. By running the problem through the linear programming package a second time with the adjusted cost function, we identify a subset of the cells that were chosen for suppression in the first run that still offers sufficient protection if suppressed. **Our** objective function is lowered if only a subset of the values chosen for suppression in the first run really need to be suppressed. We will choose to suppress as complements all cells with positive plus or minus variables from the second solution.

A third method of improving the linear programming technique of choosing complementary suppressions attempts to decrease the amount of total value suppressed in tables with more than one suppression. Because we process only one primary suppression at a time, we often create patterns of complementary suppressions for a table that are not optimal. Consider the example below where the two primaries are highlighted. Say that the k values for both primaries are 150.

1000	150	500	300	1950
150	150	500	500	1300
500	500	150	150	1300
300	500	150	**1000**	1950
1950	1300	1300	1950	6500

Processing the two primary suppressions separately with the cost function as defined in the Mathematical Formulation Section would result in the following final table.

P	**C**	500	300	1950
C	**C**	500	500	1300
500	500	**C**	**C**	1300
300	500	**C**	**1000**	1950
1950	1300	1300	1950	6500

The complementary suppressions have a total a value of

150 + 150 + 150 + 150 + 150 + 150 = 900

in this table. Another sufficient complementary suppression pattern for this table is as follows.

P	150	500	**C**	1950
150	150	500	500	1300
500	500	150	150	1300
C	500	150	**P**	1950
1950	1300	1300	1950	6500

Here the complementary suppressions have total value of

$$300 + 300 = 600.$$

Therefore, we would prefer the second complementary suppression pattern.

In order to encourage better overall complementary suppression patterns for tables with more than one primary, we can change the costs in the objective function. The Bureau is currently testing several methods of adjusting these costs. The idea behind most of the methods is to lower the costs of cells that are in rows **and** columns that have only one primary suppression, such as the cells with value 300 in the above table. Lowering the costs of these cells would increase their chance of being chosen as a complement for one primary and used again to provide protection for primaries processed after that.

VI. References

Cox, Lawrence H. (1980), "Suppression Methodology and Statistical Disclosure Control," Journal of the American Statistical Association, Volume 75, Number 370, Theory and Methods Section, American Statistical Association, Washington, D.C.

Cox, Lawrence H., Fagan, James T., Greenberg, Brian V., and Hemmig, Robert (1986), "Research at the Census Bureau into Disclosure Avoidance Techniques for Tabular Data," Proceedings of the Section on Survey Research Methods, American Statistical Association, Washington, D.C., pp 388-393.

Kelly, James P., Golden, Bruce L., and Assad, Arjang A. (1990), "Cell Suppression: Disclosure Protection for Sensitive Tabular Data," Working Paper Series MS/S 90-001, University of Maryland, College Park, Maryland.

VII. Appendix

LINDO PROGRAM

```
MIN     12 D121 + 12 D122 + 5 D131 + 5 D132 + 250 D141 + 250 D142
     + 365 D151 + 365 D152 + 12 D211 + 12 D212 + 12 D221 + 12
     D222 + 5 D231 + 5 D232 + 5 D241 + 5 D242 + 30 D251 + 30 D252
    + 40 D311 + 40 D312 + 200 D321 + 200 D322 + 90 D331 + 90
    D332 + 300 D341 + 300 D342 + 630 D351 + 630 D352 + 5 D411 +
     5 D412 + 70 D421 + 70 D422 + 50 D431 + 50 D432 + 5 D441 + 5
     D442 + 130 D451 + 130 D452 + 155 D511 + 155 D512 + 290 D521
     + 290 D522 + 150 D531 + 150 D532 + 560 D541 + 560 D542 +
     1155 D551 + 1155 D552

SUBJECT TO
        2)   D121 - D122 + D131 - D132 + D141 - D142 + D151 - D152
            + D111 - D112 =    0
        3)   D211 - D212 + D221 - D222 + D231 - D232 + D241 - D242
            + D251 - D252 =    0
        4)   D311 - D312 + D321 - D322 + D331 - D332 + D341 - D342
            + D351 - D352 =    0
        5)   D411 - D412 + D421 - D422 + D431 - D432 + D441 - D442
            + D451 - D452 =    0
        6)   D511 - D512 + D521 - D522 + D531 - D532 + D541 - D542
            + D551 - D552 =    0
        7)   D211 - D212 + D311 - D312 + D411 - D412 + D511 - D512
            + D111 - D112 =    0
        8)   D121 - D122 + D221 - D222 + D321 - D322 + D421 - D422
            + D521 - D522 =    0
        9)   D131 - D132 + D231 - D232 + D331 - D332 + D431 - D432
            + D531 - D532 =    0
       10)   D141 - D142 + D241 - D242 + D341 - D342 + D441 - D442
            + D541 - D542 =    0
       11)   D151 - D152 + D251 - D252 + D351 - D352 + D451 - D452
            + D551 - D552 =    0
```

```
 12)    D121  <=   12
 13)    D122  <=   12
 14)    D131  <=   5
 15)    D132  <=   5
 16)    D141  <=   250
 17)    D142  <=   250
 18)    D151  <=   365
 19)    D152  <=   365
 20)    D211  <=   12
 21)    D212  <=   12
 22)    D221  <=   12
 23)    D222  <=   12
 24)    D231  <=   5
 25)    D232  <=   5
 26)    D241  <=   5
 27)    D242  <=   5
 28)    D251  <=   30
 29)    D252  <=   30
 30)    D311  <=   40
 31)    D312  <=   40
 32)    D321  <=   200
 33)    D322  <=   200
 34)    D331  <=   90
 35)    D332  <=   90
 36)    D341  <=   300
 37)    D342  <=   300
 38)    D351  <=   630
 39)    D352  <=   630
 40)    D411  <=   5
 41)    D412  <=   5
 42)    D421  <=   70
 43)    D422  <=   70
 44)    D431  <=   50
 45)    D432  <=   50
 46)    D441  <=   5
 47)    D442  <=   5
 48)    D451  <=   130
 49)    D452  <=   130
 50)    D511  <=   155
 51)    D512  <=   155
 52)    D521  <=   290
 53)    D522  <=   290
 54)    D531  <=   150
 55)    D532  <=   150
 56)    D541  <=   560
 57)    D542  <=   560
 58)    D551  <=   1155
 59)    D552  <=   1155
 60)    D111  =    15
 61)    D112  =    0
END
```

LINDO SOLUTION

LP OPTIMUM FOUND AT STEP 14

OBJECTIVE FUNCTION VALUE

1) 479.00000

VARIABLE	VALUE	REDUCED COST
D121	.000000	24.000000
D122	10.000000	.000000
D131	.000000	10.000000
D132	5.000000	.000000
D141	.000000	279.000000
D142	.000000	221.000000
D151	.000000	365.000000
D152	.000000	365.000000
D211	.000000	27.000000
D212	12.000000	.000000
D221	10.000000	.000000
D222	.000000	24.000000
D231	5.000000	.000000
D232	.000000	24.000000
D241	.000000	10.000000
D242	3.000000	.000000
D251	.000000	6.000000
D252	.000000	54.000000
D311	.000000	80.000000
D312	.000000	.000000
D321	.000000	213.000000
D322	.000000	187.000000
D331	.000000	96.000000
D332	.000000	84.000000
D341	.000000	330.000000
D342	.000000	270.000000
D351	.000000	631.000000
D352	.000000	629.000000
D411	.000000	10.000000
D412	3.000000	.000000
D421	.000000	48.000000
D422	.000000	92.000000
D431	.000000	21.000000
D432	.000000	79.000000
D441	3.000000	.000000
D442	.000000	10.000000
D451	.000000	96.000000
D452	.000000	164.000000
D511	.000000	310.000000
D512	.000000	.000000
D521	.000000	418.000000
D522	.000000	162.000000

D531	.000000	271.000000
D532	.000000	29.000000
D541	.000000	705.000000
D542	.000000	415.000000
D551	.000000	1271.000000
D552	.000000	1039.000000
D111	15.000000	.000000
D112	.000000	.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	.000000	.000000
3)	.000000	-24.000000
4)	.000000	1.000000
5)	.000000	-34.000000
6)	.000000	116.000000
7)	.000000	39.000000
8)	.000000	12.000000
9)	.000000	5.000000
10)	.000000	29.000000
11)	.000000	.000000
12)	12.000000	.000000
13)	2.000000	.000000
14)	5.000000	.000000
15)	.000000	.000000
16)	250.000000	.000000
17)	250.000000	.000000
18)	365.000000	.000000
19)	365.000000	.000000
20)	12.000000	.000000
21)	.000000	3.000000
22)	2.000000	.000000
23)	12.000000	.000000
24)	.000000	14.000000
25)	5.000000	.000000
26)	5.000000	.000000
27)	2.000000	.000000
28)	30.000000	.000000
29)	30.000000	.000000
30)	40.000000	.000000
31)	40.000000	.000000
32)	200.000000	.000000
33)	200.000000	.000000
34)	90.000000	.000000
35)	90.000000	.000000
36)	300.000000	.000000
37)	300.000000	.000000
38)	630.000000	.000000
39)	630.000000	.000000
40)	5.000000	.000000
41)	2.000000	.000000
42)	70.000000	.000000
43)	70.000000	.000000
44)	50.000000	.000000
45)	50.000000	.000000
46)	2.000000	.000000
47)	5.000000	.000000
48)	130.000000	.000000
49)	130.000000	.000000
50)	155.000000	.000000
51)	155.000000	.000000
52)	290.000000	.000000

53)	290.000000	.000000
54)	150.000000	.000000
55)	150.000000	.000000
56)	560.000000	.000000
57)	560.000000	.000000
58)	1155.000000	.000000
59)	1155.000000	.000000
60)	.000000	-39.000000
61)	.000000	39.000000

NO. ITERATIONS= 14

RANGES IN WHICH THE BASIS IS UNCHANGED:

VARIABLE	OBJ COEFFICIENT RANGES CURRENT COEF	ALLOWABLE INCREASE	ALLOWABLE DECREASE
D121	12.000000	INFINITY	24.000000
D122	12.000000	6.000000	14.000000
D131	5.000000	INFINITY	10.000000
D132	5.000000	14.000000	10.000000
D141	250.000000	INFINITY	279.000000
D142	250.000000	INFINITY	221.000000
D151	365.000000	INFINITY	365.000000
D152	365.000000	INFINITY	365.000000
D211	12.000000	INFINITY	27.000000
D212	12.000000	3.000000	INFINITY
D221	12.000000	6.000000	14.000000
D222	12.000000	INFINITY	24.000000
D231	5.000000	14.000000	INFINITY
D232	5.000000	INFINITY	24.000000
D241	5.000000	INFINITY	10.000000
D242	5.000000	21.000000	3.000000
D251	30.000000	INFINITY	6.000000
D252	30.000000	INFINITY	54.000000
D311	40.000000	INFINITY	80.000000
D312	40.000000	84.000000	80.000000
D321	200.000000	INFINITY	213.000000
D322	200.000000	INFINITY	187.000000
D331	90.000000	INFINITY	96.000000
D332	90.000000	INFINITY	84.000000
D341	300.000000	INFINITY	330.000000
D342	300.000000	INFINITY	270.000000
D351	630.000000	INFINITY	631.000000
D352	630.000000	INFINITY	629.000000
D411	5.000000	INFINITY	10.000000
D412	5.000000	96.000000	3.000000
D421	70.000000	INFINITY	48.000000
D422	70.000000	INFINITY	92.000000
D431	50.000000	INFINITY	21.000000
D432	50.000000	INFINITY	79.000000
D441	5.000000	21.000000	3.000000
D442	5.000000	INFINITY	10.000000
D451	130.000000	INFINITY	96.000000
D452	130.000000	INFINITY	164.000000
D511	155.000000	INFINITY	310.000000
D512	155.000000	29.000000	271.000000
D521	290.000000	INFINITY	418.000000
D522	290.000000	INFINITY	162.000000
D531	150.000000	INFINITY	271.000000
D532	150.000000	INFINITY	29.000000
D541	560.000000	INFINITY	705.000000
D542	560.000000	INFINITY	415.000000
D551	1155.000000	INFINITY	1271.000000

D552	1155.000000	INFINITY	1039.000000
D111	.000000	INFINITY	INFINITY
D112	.000000	INFINITY	INFINITY

RIGHTHAND SIDE RANGES

ROW	CURRENT RHS	ALLOWABLE INCREASE	ALLOWABLE DECREASE
2	.000000	.000000	.000000
3	.000000	.000000	.000000
4	.000000	.000000	.000000
5	.000000	.000000	.000000
6	.000000	.000000	.000000
7	.000000	.000000	.000000
8	.000000	.000000	.000000
9	.000000	.000000	.000000
10	.000000	.000000	.000000
11	.000000	.000000	.000000
12	12.000000	INFINITY	12.000000
13	12.000000	INFINITY	2.000000
14	5.000000	INFINITY	5.000000
15	5.000000	INFINITY	.000000
16	250.000000	INFINITY	250.000000
17	250.000000	INFINITY	250.000000
18	365.000000	INFINITY	365.000000
19	365.000000	INFINITY	365.000000
20	12.000000	INFINITY	12.000000
21	12.000000	3.000000	2.000000
22	12.000000	INFINITY	2.000000
23	12.000000	INFINITY	12.000000
24	5.000000	.000000	2.000000
25	5.000000	INFINITY	5.000000
26	5.000000	INFINITY	5.000000
27	5.000000	INFINITY	2.000000
28	30.000000	INFINITY	30.000000
29	30.000000	INFINITY	30.000000
30	40.000000	INFINITY	40.000000
31	40.000000	INFINITY	40.000000
32	200.000000	INFINITY	200.000000
33	200.000000	INFINITY	200.000000
34	90.000000	INFINITY	90.000000
35	90.000000	INFINITY	90.000000
36	300.000000	INFINITY	300.000000
37	300.000000	INFINITY	300.000000
38	630.000000	INFINITY	630.000000
39	630.000000	INFINITY	630.000000
40	5.000000	INFINITY	5.000000
41	5.000000	INFINITY	2.000000
42	70.000000	INFINITY	70.000000
43	70.000000	INFINITY	70.000000
44	50.000000	INFINITY	50.000000
45	50.000000	INFINITY	50.000000
46	5.000000	INFINITY	2.000000
47	5.000000	INFINITY	5.000000
48	130.000000	INFINITY	130.000000
49	130.000000	INFINITY	130.000000
50	155.000000	INFINITY	155.000000

51	155.000000	INFINITY	155.000000
52	290.000000	INFINITY	290.000000
53	290.000000	INFINITY	290.000000
54	150.000000	INFINITY	150.000000
55	150.000000	INFINITY	150.000000
56	560.000000	INFINITY	560.000000
57	560.000000	INFINITY	560.000000
58	1155.000000	INFINITY	1155.000000
59	1155.000000	INFINITY	1155.000000
60	15.000000	2.000000	3.000000
61	.000000	3.000000	.000000