Using Mathematical Networks for Disclosure Limitation

Lawrence H. Cox, U.S. EPA

Complementary cell suppression is a technique for limiting statistical disclosure in data presented in tabular form. *Cell suppression* involves removing from publication the values of all cells causing direct disclosure of confidential data (*disclosure cells*), together with sufficiently many nondisclosure cells (*complementary cells*), to ensure that a third party cannot discover confidential respondent data by manipulating linear relationships between released and suppressed values. The challenge is to select complementary suppressions that provide sufficient disclosure protection while minimizing information lost due to suppression. This paper summarizes methods for complementary suppression (using mathematical networks) based on Cox (1995, 1996). A mathematical network is a specialized linear program defined over a mathematical graph. Networks are widely used for a variety of applications, and standard network optimization software is available. Network methods offer new theoretical and practical advantages.

Consider Table 1. Assume each cell in **boldface** is a disclosure cell **(I, J)** and for purposes here is assigned a disclosure interval of width 50-percent of its value V(I, J). Complementary cells must be selected to ensure that in the final table: $10 \le V(1, 1), V(2, 3), V(3, 4) \le 30$, and $5 \le V(4, 4) \le 15$.

Table 1

20	10	20	10	20	80
10	10	**20**	5	15	60
40	10	10	**20**	10	90
5	5	15	**10**	5	40
75	35	65	45	50	270

■ The Cell Suppression Problem

A denotes a single two-way table, comprising **m** internal rows and **n** internal columns. A contains (m+1)(n+1) entries: mn *internal entries* $\mathbf{a}_{ij}$, $1 \le i \le m$, $1 \le j \le n$, m *row totals* $\mathbf{a}_{i,n+1}$, n *column totals* $\mathbf{a}_{m+1,j}$, and the table *grand total* $\mathbf{a}_{m+1,n+1}$. A is a *positive table* if $a_{ij} \ge 0$; otherwise, A is a *general table.*

Under cell suppression, each disclosure cell **(I, J)** is suppressed from publication, together with sufficiently many complementary cells, to ensure that derived interval estimates **V(I, J)** of a_{IJ} are not narrow. Each disclosure cell is assigned a continuous range of values considered too narrow, called the *disclosure interval.* Here, the disclosure interval for (I, J) will be the symmetric open interval $(a_{IJ} - p_{IJ}, a_{IJ} + p_{IJ})$. $\mathbf{p}_{IJ} \ge 0$ is called the (symmetric) *protection limit* for (I, J).

This characterizes *interval disclosure* where the value of a disclosure cell is protected to within a continuous interval. For *exact disclosure,* only the precise value of the cell is protected. Exact disclosure is typically used for frequency counts and $p_{IJ} = 1$. The U.S. Economic Censuses involve interval disclosure; the U.S. Census of Agriculture uses exact disclosure.

Most methods are *single-cell* methods--they provide sufficient disclosure protection to a single suppressed cell at a time, and they protect the entire table by applying the method iteratively. The main requirement is to provide sufficient disclosure protection for the target cell. The second is to incur minimum information loss, usually measured by the number of suppressions or the total value suppressed. All problems considered here admit at least one *feasible solution*--suppress all entries in the table. This solution is undesirable but assures that our procedures converge. Potential disclosure in cell combinations requires additional methods not discussed here.

■ Mathematical Networks

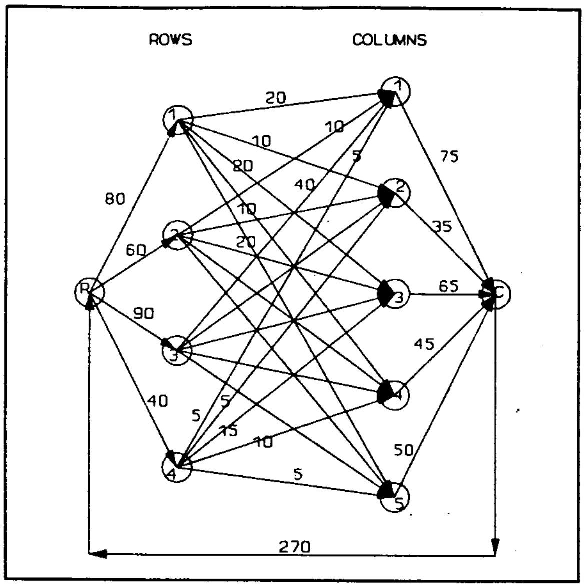
A mathematical network is a linear program uniquely suited to two-way tables. A *mathematical network* consists of a set of objects called *nodes,* together with a set of objects called *arcs* defined between ordered pairs of nodes. Nodes are denoted by letters such as **P** and **Q** and represented graphically as points. Arcs are denoted by ordered pairs of distinct nodes **(P, Q)** and **(Q, P)** and represented graphically as arrows from the first node to the second node of the pair. An arbitrary but consistent notion of direction between nodes is established, and arcs oriented in that direction are called *positive*; arcs in the opposite direction are called *negative.*

Networks represent *aggregation* relations. The assignment of quantities within a network involves *network flows* between nodes along arcs in a prescribed direction. Each node is assigned a *node requirement.* A positive requirement indicates the amount of net outflow required at the node (sum of flows along arcs directed out of the node minus sum of flows directed into the node); a negative requirement indicates net inflow required; and a zero node requirement indicates balanced in- and outflow. Row and column nodes are denoted **i** and **j**, and flows are denoted $\mathbf{x}_{ij+}$ and $\mathbf{x}_{ij-}$, relative to flow direction.

B denotes the *node-arc incidence matrix* of the network: **B** contains one row for each node and one column for each arc. The entry in the arc-column for the from-node of the arc equals +1; that for the to-node equals -1; others in the arc-column equal 0. **x** denotes the column vector of variables corresponding to the arc flows. **R** denotes the row vector of node requirements. The linear constraint system of the network is: **Bx = R**, $\mathbf{x} \geq \mathbf{0}$. The **x**-values can be restricted by (upper) *capacity constraints* $\mathbf{x} \leq \mathbf{u}$, for **u** a column vector of nonnegative upper limits on individual flows. The relationship between networks and two-way tables is illustrated in Figure 1, the network for Table 1. Oppositely directed arcs are not drawn.

A *network optimization problem* (*N*, **c**) consists of a network *N* with a cost function **c** to be minimized subject to the linear constraints of the network structure. The (upper) *capacitated network optimization problem* is: **min cx, Bx = R,** $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$.

Figure 1



Assume *N* is the network representation of a positive table A. Given arc (I, J, +) of *N*, a *circuit* γ in *N* containing this arc is a sequence of distinct arcs of *N* satisfying: the to-node of each arc equals the from-node of the successive arc; and the to-node of the last arc equals the J-column node. If successive arcs in γ are in opposite directions, it is an *alternating cycle.* For

$$g(\gamma) = \min\{a_{ij}: (i, j, + \text{ or } -) \in \gamma\},$$

we see that there exist flows of up to **g(γ)** units subject to $\mathbf{x} \geq 0$: add **g(γ)** to each positive arc on γ and subtract **g(γ)** from each negative arc. By reversing arc directions, flow up to **g(γ)** units in the reverse direction is possible. Thus, V(I, J) can assume any value in the interval

$$[a_{IJ} - g(\gamma),\ a_{IJ} + g(\gamma)]\ ;$$

intervals of equal width hold for other cells on γ. An alternating cycle is: (1, 1, +), (1, 4, -), (3, 4, +), (3, 3, -), (2, 3, +),(2, 1, -). Flow of 10 units in either direction is possible.

Observation. A disclosure cell (I, J) with protection limit p_{IJ} is protected if there exists an alternating cycle γ comprising only suppressed cells that contain (I, J) and satisfy $g(\gamma) \geq p_{IJ}$. Networks enjoy a property crucial to our development.

Integrality Property. If **R** and **u** and integer and **c**

are nonconstant, then any optimal solution **x** is integer.

We use the Integrality Property powerfully: If $\mathbf{u} \leq 1$ is integer-valued, then both x_{ij+}, $x_{ij-} = 0$ or 1 for all (i, j). This enables using computationally efficient, continuous network optimization to solve a dichotomous decision problem--complementary suppression. $\mathbf{x}_{ij*}$ denotes either x_{ij+} or x_{ij-}, or both.

Dichotomous Network Model for Cell Suppression

Network for Single-Cell Suppression in a Single Table

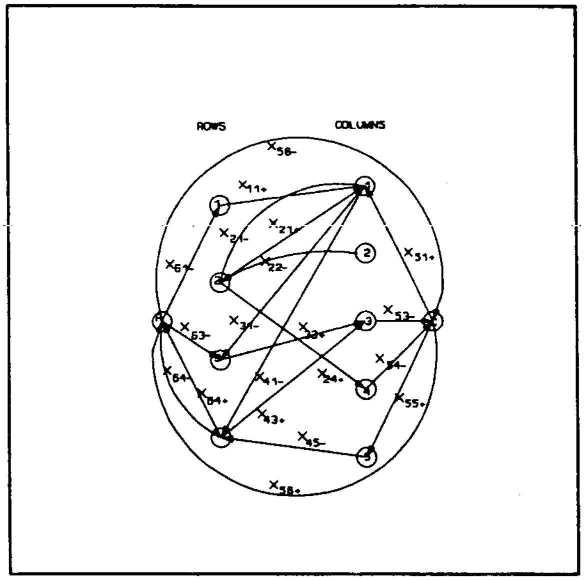
The cell suppression problem for a single table subject to minimum-number-of-complementary-suppressions or minimum-total-value-suppressed can be modelled as a dichotomous network. **S** denotes the set of previously-suppressed cells, **#S** denotes the number of cells in S, **(I, J)** ∈ S denotes the *target cell*--the suppressed cell being disclosure-protected at the current iteration. $U \subseteq S$ denotes the set of table cells that remain unprotected. $\bar{S} = S \sim (I, J)$. The network *N* is run once for each unprotected suppressed cell (I, J) $\in U$. At each iteration, (I, J) and perhaps additional unprotected suppressed cells $(I^*, J^*) \in U$ are protected by creation of an alternating cycle containing (I, J). The network *N* for Table 1 and target cell (1, 1) is illustrated in Figure 2.

Nodes. There are m+n+2 nodes. The first m nodes correspond to the m internal rows of A, the next n nodes correspond to the n internal columns of A, and the last 2 nodes correspond to the column of row totals and the row of column totals adding to the grand total.

Arcs. There are (m+1)(n+1) positive arcs: one from each row node to each column node, one from each row node to the first grand total node, one from the second grand total node to each column node, one from the second grand total node to the first grand total node. Similarly, between the same node pairs, there are (m+1)(n+1) oppositely directed negative arcs. Flows are denoted by $\mathbf{x}_{ij+}$ and $\mathbf{x}_{ij-}$, $1 \leq i \leq m+1$, $1 \leq j \leq n+1$.

Node Requirements. All are zero.

Figure 2



Arc Capacities. The capacities of x_{IJ+} and x_{IJ-} are $u_{IJ+} = 1$ and $u_{IJ-} = 0$. If $a_{ij} = 0$, then $u_{ij*} = 0$. Unless stated otherwise, $u_{ij*} = 1$ otherwise.

Arc Costs. $c_{IJ+} = -(c_0 + 1)$, for

$$c_0 = \sum_{(i,j) \neq (I,J)} (c_{ij+} + c_{ij-})$$

. $c_{IJ-} = 1$. For

$$(I^*, J^*) \in \bar{S},\ c_{I^*J^*+} = c_{I^*J^*-} = 1 .$$

For (i, j) ∉ S, arc costs depend upon the problem type but are subject to c_{ij+}, $c_{ij-} \geq$ #S.

The Integrality Property and the {0, 1} capacities ensure that each arc flow in the optimal solution satisfies $x_{ij*} = 0$ or 1. This permits the network optimization and complementary cell suppression problems to be connected by the following rule.

Cell Suppression Rule. Suppress cell (i, j) if $x_{ij+} = 1$ or $x_{ij-} = 1$ in the optimal solution.

All flows x_{ij*} are nonnegative. The large negative arc cost c_{IJ+} forces $x_{IJ+} = 1$, avoiding the trivial solution $\mathbf{x} = \mathbf{0}$ and forcing suppression of cell (I, J). The zero node requirements impose the constraints

$$\sum_{j=1}^{n} x_{ij+} = \sum_{j=1}^{n} x_{ij-} \ , \ 1 \le i \le m+1$$

$$\sum_{i=1}^{m} x_{ij+} = \sum_{i=1}^{m} x_{ij-} \ , \ 1 \le j \le n+1$$

These conditions and $x_{IJ+} = 1$ ensure that an optimal solution to (*N*, c) is an alternating cycle γ containing (I, J). The arc capacity $u_{IJ-} = 0$ forces $x_{IJ-} = 0$, ensuring that the cycle is non-trivial (i.e., $k \geq 4$). This cycle is the *protection cycle* for the target cell (I, J). The only negative arc cost is c_{IJ+}. The next smallest costs are $c_{I^*J^*+} = c_{I^*J^*-} = 1$ for previously-suppressed cells $(I^*, J^*) \in \mathcal{S}$. Together with c_{ij+}, $c_{ij-} \geq \#S$ for $(i, j) \notin S$, this encourages the optimization to select previously-suppressed cells as complementary suppressions for the target cell (I, J). As $c_{ij+}, c_{ij-} > 0$ for $(i, j) \neq (I, J)$, trivial subcycles (i.e., $x_{ij+} = x_{ij-} = 1$) are avoided, ensuring no *superfluous suppressions*.

Optimizing Single-Cell Suppression Under Minimum-Number-of-Complementary-Suppressions

Arc Costs: General Table. $c_{ij+} = c_{ij-} = \#S$ for all $(i, j) \notin S$.

Thus, the total cost c(x) in an optimal solution $\{x^*\}$ satisfies: $c(x^*) = (\#S)r + s - (c_0 + 1)$, where **r** equals the minimum number of complementary suppressions needed, and **s** equals the number of previously-suppressed cells in the optimal protection-cycle. This is guaranteed by the way the arc costs were *stratified*: the large negative cost forces suppression of (I, J); larger positive costs ensure r is minimized; and unit costs ensure that any previously-suppressed cell that can be used will be used, but only once. If r = 0, then only previously-suppressed cells are needed and the network has *verified* protection.

The network optimization (*N*, c) solves the cell suppression problem for general tables optimally under minimum-number-of-suppressions as the solution involves precisely r complementary suppressions.

There are typically many optimal solutions under the minimum-number-of-suppressions criterion. One can use the cost function to select one of minimum total value (see Cox 1995 for details and cost functions **c** and **c'**).

Exact disclosure in positive tables

(*N*, c) optimally solves the complementary cell suppression problem for exact disclosure in positive tables under the minimum-number-of-complementary-suppressions criterion. (*N*, c') optimally solves this problem under the two-stage criterion. Moreover, the protection-cycle for target cell (I, J) protects all complementary suppressions made at the current iteration, thus avoiding unnecessary iterations.

Interval disclosure in positive tables

(*N*, c') incorporates the cell value a_{ij} into the optimization criterion as a refinement of (*N*, c) for the case of exact disclosure in positive tables. A similar but refined approach is needed for interval disclosure.

Optimal methods are not available for interval disclosure in positive tables, nor for the minimum-total-value-suppressed criterion in general and positive tables. This stems from two factors. The first is the difficulty of incorporating both dichotomous decision variables and continuous variables representing protection levels into a single, computationally efficient linear programming formulation. The second is that the complementary cell suppression problem under the minimum-total-value-suppressed criterion is *NP hard*, suggesting that the existence of a provably efficient (polynomial time) algorithm is unlikely. The methods presented are heuristic methods. Interval disclosure in positive tables requires that the cycle(s) through target cell (I, J) permit a flow of at least p_{IJ} units in each direction. Depending upon the a_{ij}, this may require more than one cycle through (I, J). The iterative step in the heuristic procedure attempts to provide sufficient protection along a single cycle, whenever possible, as follows.

Arc Capacities And Costs. Modify *N* to create *N'* satisfying $u_{ij*} = 0$ whenever $u_{ij*} = 1$ in *N* and $a_{ij} < p_{IJ}$. If

$(i,\ j)\in\overline{S}$, $c_{ij\bullet}$=1; if $(i,\ j)\notin S$, $c_{ij\bullet}$=#S.

IP(I, J) is an heuristic procedure for complementary disclosure for (I, J) and *cell reuse* under interval disclosure in positive tables.

IP(I,J)

INITIALIZE. Set $b_{ij*}=a_{ij}$, $q_{IJ}=p_{IJ}$, $S^* = \varnothing$. @ denotes + or -, and & the opposite sign. In N' and N, for $(i,j)\notin S$: $p_{ij}=0$; if i=m+1 or j=n+1, $u_{ij\bullet}=0$.

STEP 1. Substitute q_{IJ} for p_{IJ} in N'. Attempt an optimal solution (x, γ) to (N',c).

STEP 1a. If none exists, attempt an optimal solution (x, γ) to (N,c).

STEP 1b. If none exists, for (i,j) ≠ (I,J) and i=m+1 or j=n+1, set $u_{ij*}=1$ in N whenever $a_{ij}\geq q_{IJ}$, and compute an optimal solution (x, γ) to (N,c). (Reset $u_{ij*}=0$ after Step 3).

STEP 2. Compute

$$g(\gamma) = \min\{b_{i,j,@}: (i,j,@)\in\gamma\} \quad .$$

STEP 3. For $(i,j,@)\in\gamma,\ (i,j)\neq(I,J)$:

Replace $b_{ij@}$ by $b_{ij@}$ - $g(\gamma)$ and $b_{ij\&}$ by $b_{ij\&}$ + $g(\gamma)$. If $b_{ij@}=0$, set $u_{ij@}=0$. If $u_{ij\&}=0$, set $u_{ij\&}=1$.

Set $p_{ij} = \max\{p_{ij},\ a_{ij}-b_{ij@},\ a_{ij}-b_{ij\&}\}$.

If $(i,j)\notin S\cup S^*$, adjoin (i,j) to S^* .

STEP 4. Replace q_{IJ} by q_{IJ} - g (γ).

If q_{IJ}>0, go to STEP 1.

STEP 5. (Restore superfluous suppressions.) For $(i,j)\in S^*$ with $b_{ij+}=b_{ij-}$, remove (i,j) from S^* and set $p_{ij}=0$.

STEP 6. Replace S by $S\cup S^*$.

STEP 7. END.

Optimizing Single-Cell Suppression Under Minimum-Total-Value-Suppressed

A sufficient complementary cell suppression pattern for (I, J) that minimizes total-value-suppressed is obtained from the network optimization (*N*, c), where

$c_{ij\bullet}$ = 1 if $(i,j)\in\overline{S}$; $c_{ij\bullet}$ = #S + a_{ij} if $(i,j)\notin S$, and $c_{ij\bullet}$ is large.

Large-scale linear programming implementations of heuristic procedures are due to the U.S. Census Bureau and Statistics Canada. The Census method is based on network optimization using flow variables $\mathbf{z}_{ij*}$ representing the disclosure-protection provided by suppressing cell (i, j); arc costs $\mathbf{d}_{ij*}$ are based upon the cell value a_{ij}. The cell is suppressed if either z_{ij+}, z_{ij-} > 0 in the optimal solution. Statistics Canada uses general linear programming and the same flow variables z_{ij*} and decision rule with arc costs d_{ij*} based upon $\log(1+a_{ij})$. Both seek minimum-total-value-suppressed. However, in each case, the cost function $\boldsymbol{d(z)} = \sum_{i,j} \boldsymbol{d_{ij\bullet} z_{ij\bullet}}$ is a poor surrogate for actual minimum-total-value-suppressed $c(x) = \sum_{i,j} a_{ij\bullet}x_{ij\bullet}$, whereas c(x) appears explicitly in our cost formulation, a theoretical improvement. Another theoretical improvement is that our method is equivalent to a *minimum path* solution.

Multiple-Cell Complementary Suppression

A method that provides disclosure protection to all suppressed cells in a single step is a *multiple-cell complementary suppression* method. The existence of efficient multiple-cell methods is clouded by NP hardness results. The method below demonstrates that progress is possible.

Problem. Given a single two-way table A and a set of suppressions S, select a minimal set of complementary suppressions so that each row and column of the table containing suppression(s) contains at least two suppressions.

The *Problem* is a necessary condition to the minimum-number-of-suppressions problem, but is not a sufficient condition in that there may exist patterns satisfying the two-or-more-suppressions-per-affected-row-or-column condition for which all suppressed cells are not contained in a cycle (i.e., fail the *cycle condition*). The number of patterns satisfying the two-or-more condition for a given table is computable and, in general, is large, and, in practice, most two-or-more patterns also satisfy the cycle condition or can be augmented.

Theorem. Let A be a general two-way table with suppressions under the minimum-number-of-complementary-suppressions criterion. Let **m'** (respectively, **m"**) denote the number of rows of A containing suppressions (respectively, requiring complementary suppression), and define **n'** (respectively, **n"**) similarly. Assume m" $\geq$ n" and m" $\geq$ 1. If max {m', n'} = 1, then the *Problem* can be solved by three complementary suppressions. Otherwise, m" complementary suppressions are sufficient.

The *Problem* can be formulated as a network optimization problem **(*M*, e)**, as follows. For clarity of presentation, it suffices to assume that suppression is limited to the internal entries of the table. There are two degenerate cases, each solvable by methods established earlier: 1) max {m', n'} = 1 and 2) n" = 0 and n' = 1. The first can be solved using the network model (*N*, c). The second, which occurs when all suppressions occur in a single column **J**, can be solved using (*N*, c), except $c_{iJ} = -(c_0 + 1)$ for $(i, J) \in U$. The general case, min {m', n'}>1, is solved by a network.

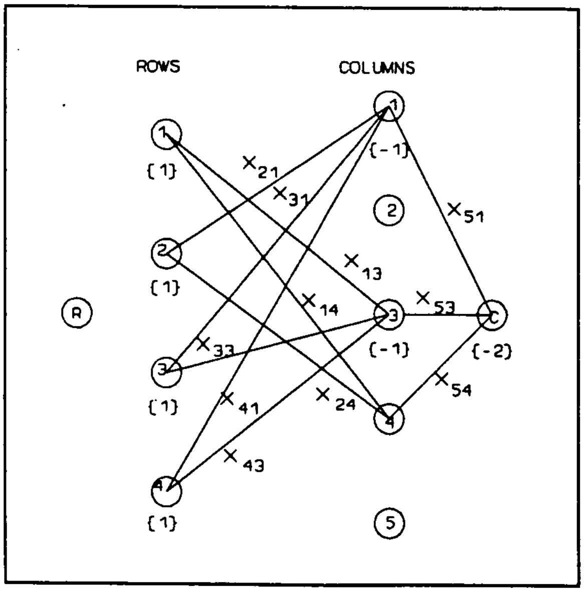
NETWORK *M*. The node set of *M* equals the node set of *N*. The arc set of *M* consists of one arc from each row node **i** corresponding to a row requiring complementary suppression to each column node **j** corresponding to a column containing suppressions (flows denoted $\mathbf{x}_{ij}$), and one arc from each of these column nodes to the second grand total node (flows denoted $\mathbf{x}_{m+1,j}$). Arc capacities are: $u_{m+1,j} = \infty$; $u_{ij} = 0$ if $(i, j) \in S$ or if $a_{ij} = 0$; $u_{ij} = 1$ otherwise. Node requirements are zero, except: the requirement of a row node corresponding to a row requiring complementary suppression equals +1; that of a column node corresponding to a column requiring complementary suppression equals -1; and, that of the second grand total node equals **-(m" - n")**.

The node requirements ensure a total flow of m" units, one unit along each row, corresponding to the minimum number of complementary suppressions m" required to disclosure-protect a general table--one in each row requiring complementary suppression. The column node requirements ensure that each column requiring complementary suppression receives at least one complementary suppression. The distribution of flow only to columns containing suppressions ensures that new column suppression problems are not created. Figure 3 illustrates the network *M* corresponding to the complementary suppression problem for Table 1. Node requirements are denoted by { }; zero-capacity arcs have been deleted.

In general tables, any basic feasible solution to *M* that obeys the cycle condition is an optimal solution. For positive tables, additional conditions often must be imposed. These conditions are expressible in terms of arc capacities **u** and a cost function **e** for *M*. To obtain a sufficient solution in one optimization (whenever such exists), the condition

$$u_{ij} = 0 \text{ if } a_{ij} < \max\{p_{IJ}: (I, J) \in S\}$$

Figure 3



is imposed. To include a secondary optimum (e.g., minimum-total-value-suppressed subject to minimum-number-of-complementary-suppressions), costs such as $e_{ij} = a_{ij}$ can be used. (M, e) can be used to generate potential multiple-cell solutions in one optimization step.

Discussion

Our methods offer theoretical advantages over methods in use (formulating a dichotomous decision problem explicitly as a linear optimization problem, use of minimum path), and practical advantages (computational efficiency, reliance on standard methods and software, flexibility in use). Stratified cost functions enable finding optimal solutions and combining multiple optimizations into one optimization.

An optimal solution to the cell suppression problem of Table 1 consists of the following complementary suppressions: (1, 4), (2, 1), (3, 3), and (4, 1). This is a minimum-number-of-complementary-suppressions solution, which provides sufficient protection to all table cells. It is also a minimum-total-value-suppressed solution. This "dual-optimum" is unlikely to occur in practice.

References

Cox, Lawrence (1995), "Network Models for Complementary Cell Suppression," *Journal of the American Statistical Association* **90**, pp. 432, 1453-1462.

_____(1996), "Addendum," *Journal of the American Statistical Association* **91**, pp. 436, 1757.