Overview of Cell Suppression Methods

Ruiyi Zhang* Lu Chen † Yang Cheng ‡ Michael Jacobsen §

Abstract

Statistical agencies widely use cell suppression methods for economic censuses and establishment surveys to protect sensitive tabular data from disclosure to the public. The goal is to reduce the risk of disclosure by first identifying sensitive cells as primary suppression cells and then finding additional cells as the complementary (secondary) suppression cells to protect the primary cells against an attacker. In general, the cell suppression problems (CSP) can be described as a linear programming problems. In this paper, we review the cell suppression problem, with a focus on the network flow model of two-dimensional tables as well as the heuristic solutions and the exact optimal solutions. Applications of cell suppression methods from statistical agencies are highlighted. The extension of the solutions to high-dimensional, hierarchical, and linked tables is also discussed.

Key Words: Cell suppression, Linear programming, Network flow, Optimal solution

*National Institute of Statistical Sciences and USDA National Agricultural Statistics Service, 1400 Independence Avenue, SW, Washington, DC 20250-2054. Email: Ruiyi.Zhang@usda.org

†National Institute of Statistical Sciences and USDA National Agricultural Statistics Service, 1400 Independence Avenue, SW, Washington, DC 20250-2054. Email: Lu.Chen@usda.org

‡USDA National Agricultural Statistics Service, 1400 Independence Avenue, SW, Washington, DC 20250-2054. E-mail: Yang.Cheng@usda.gov

§USDA National Agricultural Statistics Service, 1400 Independence Avenue, SW, Washington, DC 20250-2054. E-mail: Michael.Jacobsen2@usda.gov

1. Introduction

In the modern era of data ubiquity, the handling and publication of sensitive data pose profound challenges, especially for statistical agencies responsible for disseminating information to the public while also preserving confidentiality. As tabular data formats continue to be a primary means of public data presentation, quite a lot of statistical surveys or censuses still rely on techniques like cell suppression to mitigate the risk of statistical disclosure.

The United States Department of Agriculture's (USDA's) National Agricultural Statistics Service (NASS) conducts hundreds of surveys each year on issues including agricultural production, economics, demographics, and the environment. Every five years NASS also conducts the Census of Agriculture, providing the only source of uniform, comprehensive agricultural data for every county in the nation. Statistical disclosure limitation methods are applied by NASS to limit the risk of disclosure of individual information when statistics are disseminated to the public. It is also required by law to protect the confidentiality of individual information in the production of official statistics. For example, there are Title 7 of U.S. Code, Sec 2276 – Confidentiality of Information – Agricultural Title – 1985; Confidential Information Protection and Statistical Efficiency Act (CIPSEA) of 2018; Title III of the Evidence Act of 2019; Title 13, U.S. Code, Title 26, U.S. Code, and many others.

The current NASS cell suppression disclosure system was developed in the 1990s mainly based on Cox (1995). However, many newly developed cell suppression methods can improve the computational efficiencies and the protection of the data. Research at NASS is underway to improve the disclosure limitation methods. In this paper, we mainly review the work by Kelly et al. (1992), Cox (1995), Fischetti and Salazar (1999), Fischetti and González (2000), and Steel et al. (2013) for their efforts in solving the cell suppression problem for tabular data.

Kelly et al. (1992) laid the foundation for representing the two-dimensional table as a network and formulating the cell suppression problem as a mixed integer linear programming (MILP) problem. Besides, Kelly et al. (1992) proved the cell suppression problem to be an NP-hard (nondeterministic polynomial time) problem. The paper also proposed a network-based heuristic solution under the min-total-value-of-complementary-suppressions criterion. Building upon the network model and the linear optimization formulation, Cox (1995) proposed another heuristic solution for two-dimensional tables. Fischetti and Salazar (1999), on the other hand, delved into the computational aspect of the problem, introduced a new integer

linear programming model and employed the branch-and-cut algorithm to achieve exact solutions for large-scale instances of two-dimensional tables. Fischetti and González (2000) and Steel et al. (2013) extended the linear programming framework of cell suppression to three or higher-dimensional table, hierarchical tables, and linked table.

In summary, these papers illuminate various aspects of the cell suppression problem, from mathematical formulations to computational techniques, and from the heuristic solutions to the exact solutions. This review paper aims to synthesize these contributions, evaluate their limitations, and explore potential future directions for NASS to improve the current statistical disclosure system.

The paper is structured as follows. Section 2 introduces the cell suppression problems. The network model as a representation of two-dimensional tables and a general linear programming framework of the cell suppression problem are presented in Section 3. Section 4 illustrates the different formulations of the protection level constraints in literature as well as the heuristic solutions and the exact optimal solutions. The summary and some discussion of possible future research are stated in Section 5.

2. The Cell Suppression Problem

Magnitude data shown in tables, often derived from surveys or censuses of businesses, farms, institutions, etc., are generally non-negative numbers. Their distribution is typically skewed, meaning a few entities contribute large values. Disclosure limitation, in this case, is to ensure that the released data does not allow for precise estimate of the values contributed by the most significant respondents. The most commonly primary suppression rules used by statistical agencies to identify sensitive cells, referred to as *primary* cells, in the tabular data are the (n) threshold rule, (n, k) rule, and the p-percent or pq rules.

To protect the set of primary cells in a table with margin totals, the most straightforward way is to suppress them. However, the suppression of the primary cells alone can be easily attacked through the margin totals. It is therefore necessary to suppress additional cells, which are termed as *complementary* cells. The purpose of cell suppression is to guarantee that the attacker cannot get an estimation "close" enough to the true cell value. The measurement of "closeness" is provided by the protection levels, which are derived from some primary suppression rule. For example, Table 1 shows an example of a two-dimensional table along with one primary cell 1000 and its protection level 23, then the predetermined protection interval is $[1000 - 23, 1000 + 23] = [977, 1023]$. Table 2 shows one cell suppression solution. The attacker's interval

Table 1: Example of a two-dimensional table. The predetermined protection level of the primary cell 1000 is 23.

	col_1	col_2	col_3	col_4	col_T
row_1	**1000 P**	13	18	25	1056
row_2	12	10	40	50	112
row_3	17	35	15	23	90
row_4	30	28	40	200	298
row_5	27	55	20	19	121
row_T	1086	141	133	317	1086

Table 2: Cell suppression solution 1 by minimizing the total number of complementary cells of Table 1. Total complementary cell value = 255 with 3 complementary cells.

	col_1	col_2	col_3	col_4	col_T
row_1	*	13	18	*	1056
row_2	12	10	40	50	112
row_3	17	35	15	23	90
row_4	*	28	40	*	298
row_5	27	55	20	19	121
row_T	1086	141	133	317	1086

estimate of the primary cell 1000, with some simple algebraic calculation using column totals (the last column) and row totals (the last row) in Eq. (1), where the suppressed cells are denoted as x_1, x_2, x_3 and x_4, is $[800, 1025]$, which is wider than the required protection level $[977, 1023]$. Therefore, the cell suppression solution 1 in Table 2 protects the primary cell successfully.

$$\left.\begin{aligned} x_1 + 13 + 18 + x_2 &= 1056 \\ x_3 + 28 + 40 + x_4 &= 298 \\ x_1 + 12 + 17 + x_3 + 27 &= 1086 \\ x_2 + 50 + 23 + x_4 + 19 &= 317 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}\right\} \Rightarrow \left.\begin{aligned} x_1 &= 800 + x_4 \\ x_2 &= 255 - x_4 \\ x_3 &= 230 - x_4 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}\right\} \Rightarrow \left.\begin{aligned} x_1 &\in [800, 1025] \\ x_2 &\in [0, 225] \\ x_3 &\in [5, 230] \\ x_4 &\in [0, 225] \end{aligned}\right\} \tag{1}$$

We will only consider tables with non-negative entries in this paper. Let matrix $A = (a_{ij})$ with $a_{ij} \geq 0$ represent a two-dimensional table of size $(m+1)\times(n+1)$ with margin totals. For more compact expressions, we also use a vector $\vec{a} = \{a_1, a_2, \cdots, a_T\}$, where $T = (m+1) \times (n+1)$ to represent the cell values. Let $\mathcal{P} = \{(i_k, j_k), k = 1, \cdots, p\}$ be the set of primary cells, then we can officially define the protection levels

of the primary cell $(i_k, j_k), k = 1, \cdots, p$.

Definition 1. (Protection Interval) *For each primary cell* $(i_k, j_k) \in \mathcal{P}, k = 1, \cdots, p$*, the protection interval is*

$$[a_{i_k j_k} - LPL_k, a_{i_k j_k} + UPL_k],$$

where the upper protection level UPL_k *and the lower protection level* LPL_k *for the primary cell* (i_k, j_k) *are predetermined by the statistical agency with some primary suppression rule.*

In most cases, the upper and lower protection levels are equal but they also can be different. The purpose of cell suppression is to guarantee that the attacker's interval estimate of the cell value $a_{i_k j_k}$ is wider than the required protection interval $[a_{i_k j_k} - LPL_k, a_{i_k j_k} + UPL_k]$.

Table 3: Cell suppression solution 2 of Table 1. Total complementary cell value = 165 with 5 complementary cells.

	col_1	col_2	col_3	col_4	col_T
row_1	*	13	18	*	1056
row_2	12	10	40	50	112
row_3	17	*	15	*	90
row_4	30	28	40	200	298
row_5	*	*	20	19	121
row_T	1086	141	133	317	1086

Table 4: Cell suppression solution 3 by minimizing the total value of complementary cells of Table 1. Total complementary cell value = 85 with 6 complementary cells.

	col_1	col_2	col_3	col_4	col_T
row_1	*	*	*	25	1056
row_2	*	*	40	50	112
row_3	*	35	*	23	90
row_4	30	28	40	200	298
row_5	27	55	20	19	121
row_T	1086	141	133	317	1086

Table 2, 3, and 4 show three different cell suppression solutions to protect the primary cell $a_{11} = 1000$ with required protection level 23. They all protect the primary cell $a_{11} = 1000$ successfully. However, they reflect different loss of information as shown in Table 5. The measurement of information loss in cell suppression is either total number of complementary cells, total value of complementary cells, or a combination of the two criteria.

Table 5: Information loss of the three cell suppression solutions of Table 1.

	Total value of complementary cells	Total number of complementary cells
Solution 1	255	3
Solution 2	165	5
Solution 3	85	6

Table 6: A general setting of a two-dimension table of size $(m+1) \times (n+1)$.

	col_1	col_2	$\cdots$	col_n	col_T
row_1	a_{11}	a_{12}	$\cdots$	a_{1n}	$a_{1,n+1}$
row_2	a_{21}	a_{22}	$\cdots$	a_{2n}	$a_{2,n+1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
row_m	a_{m1}	a_{m2}	$\cdots$	a_{mn}	$a_{m,n+1}$
row_T	$a_{m+1,1}$	$a_{m+1,2}$	$\cdots$	$a_{m+1,n}$	$a_{m+1,n+1}$

In summary, the cell suppression problem can then be described as follows. Given a set of primary cells along with the required protection levels, the objective is to find a set of complementary cells to protect the primary cells against the attacker (make sure the attacker's interval estimate of the cell value is wider than the predetermined protection levels) while minimizing the information loss. A general setting of a $(m+1) \times (n+1)$ two-dimensional table is shown is Table 6. Let matrix $A = (a_{ij})$ with $a_{ij} \geq 0$ represent the given table with non-negative entries. The margin totals introduce the following conditions:

$$\begin{aligned} a_{i,n+1} &= \sum_{j=1}^{n} a_{ij} \quad \text{for } i = 1, \cdots, m \quad \text{row totals;} \\ a_{m+1,j} &= \sum_{i=1}^{m} a_{ij} \quad \text{for } j = 1, \cdots, n \quad \text{column totals;} \\ a_{m+1,n+1} &= \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} \quad \text{grand total.} \end{aligned} \tag{2}$$

If we put all of the cell value a_{ij} into one vector $\vec{a} = [a_1, a_2, \cdots, a_T]$, then the above equations in Eq. (2) can be written into a more compact way:

$$M\vec{a} = \vec{0}, \tag{3}$$

where M is a matrix with values of $\{-1, 0, +1\}$.

To find the complementary cells, we minimize the following objective function which represents the

information loss

$$\min \sum_{i=1}^{(m+1)} \sum_{j=1}^{(n+1)} c_{ij} x_{ij} \tag{4}$$

subject to

$$x_{ij} = 0 \text{ or } 1 \tag{5}$$

$$x_{i_k j_k} = 1, \text{ if } (i_k, j_k) \in \mathcal{P} \tag{6}$$

where x_{ij} is the indicator for suppression, and cost c_{ij} represents the information loss when we suppress cell $)i, j)$. For example, if $c_{ij} = 1$, the objective function is seeking the least total number of suppressed cells. If $c_{ij} = a_{ij}$, the objective function is seeking the least total value of suppressed cells.

Eq. (4), (5) and (6) are the basic settings of the cell suppression problem. Different papers handle the protection level constraints in different ways and we will discuss the details in Section 4.

3. The Network Model for the Cell Suppression Problem

A two-dimensional table can be naturally represented by a network (See Fig. 1). The network optimization formulation of the cell suppression problem for two-dimensional tables was first proposed in Kelly et al. (1992) and then adopted by Cox (1995) and Fischetti and Salazar (1999) as well as other literature on cell suppression. However, this elegant structure is not preserved for 3-dimensional or higher-dimensional tables, unless it can be deconstructed into a collection of independent two-dimensional subtables.

Definition 2. (Graph) *A graph is a pair $G = (V, E)$, where V is a set whose elements are called vertices, and E is a set of paired vertices, whose elements are called edges.*

In operations research, the vertices are usually called nodes and the edges are called arcs. We will follow that convention in this paper.

Definition 3. (Basic Network) *A basic network is a directed graph $D = (V, A)$ with the following properties:*

1. *Exactly one source node, where the source node is a node with zero in-degree edges.*
2. *Exactly one sink node, where the sink node is a node with zero out-degree edges.*

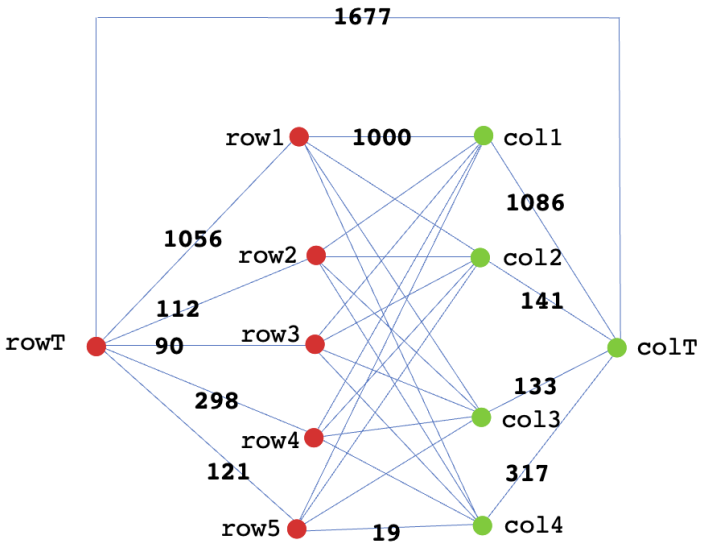

Figure 1: Network $D = (V, A)$ for Table 1.

3. *Each edge e of G has a positive capacity $q(e)$, where $q(e)$ = the maximum amount of the flow that can go through the edge e.*

Fig. 1 shows an example of the network representation of Table 1, where $rowT$ is the source node, $colT$ is the sink node, and the capacity of edge $e = (i, j)$ is q_{ij}.

Definition 4. (Flow) *Flow is an assignment f_{ij} to each edge $e = (i, j)$ of the network that satisfies:*

1. *The feasibility condition:*

$$f_{ij} \leq q_{ij} \tag{7}$$

2. *The flow conservation law: the total flow entering a node must equal the total flow leaving a node except the source node and the sink node.*

The linear system expressed in Eq. (2) and Eq. (3) which guarantees the margin totals in table $A = (a_{ij})$ is equivalent to the flow conservation law in network. For example, for the node row_1, the in-flow = 1056 and the out-flow = $[1000, 13, 18, 25]$ which are equal.

4. Solutions to the Cell Suppression Problem

Based on the common linear programming framework defined in Eq. (4), (5), (6), this section will focus on the how these papers construct the linear constraints of the protection level requirement. Recall that a successful cell suppression solution must make sure that the attacker's estimate of the primary cell i_k must be wider than the required protection interval $[a_{i_k} - LPL_k, a_{i_k} + UPL_k]$.

4.1 Heuristic Solutions to the Cell Suppression Problem

There are two main heuristic solutions proposed by Kelly et al. (1992) and Cox (1995), respectively. They have something in common:

1. They are both iterative methods, which means that they protect the primary cells sequentially, i.e., one at a time.

2. Within each iteration, a heuristic network-based solution is proposed to satisfy the protection level constraints for the primary cell.

The difference between Kelly et al. (1992) and Cox (1995) are the ways they handles the protection level constraints.

Now, consider the published Table 2 with 4 suppressed cells from the perspective of the attacker. The goal of the attacker is to infer the suppressed cell values. Assume that the *feasible values* for the suppressed cells are f_{ij}'s, and the set of suppressed cells is $\mathcal{P} \cup \mathcal{C}$ where $\mathcal{P}$ and $\mathcal{C}$ represent the set of primary cells and complementary cells, respectively. Then, the attacker would have:

$$\left.\begin{aligned} M\vec{f} &= \vec{0} \\ f_{ij} &= a_{ij}, \text{ for } (i,j) \notin \mathcal{P} \cup \mathcal{C} \\ lb_{ij} \leq f_{ij} &\leq ub_{ij}, \text{ for } (i,j) \in \mathcal{P} \cup \mathcal{C} \end{aligned}\right\}, \tag{8}$$

where $[lb_{ij}, ub_{ij}]$ represents the possible range of the cell value a_{ij} known to the attacker, usually $lb_{ij} = 0$ and $ub_{ij} = +\infty$ for tables with non-negative entries.

Assume the prespecified upper and lower protection level for the primary cell (i_k, j_k) is UPL_k and LPL_k, respectively. Next, the required protection levels are translated the into the linear constraints by introducing two sets of auxiliary continuous variables for the primary cell (i_k, j_k):

Table 7: One feasible solution for suppressed Table 2.

	col_1	col_2	col_3	col_4	col_T
row_1	**980**	13	18	**45**	1056
row_2	12	10	40	50	112
row_3	17	35	15	23	90
row_4	**50**	28	40	**180**	298
row_5	27	55	20	19	121
row_T	1086	141	133	317	1086

$$\mathbf{G}^k = \begin{bmatrix} g_{11}^k & g_{12}^k & \cdots & g_{1,n+1}^k \\ g_{21}^k & g_{22}^k & \cdots & g_{2,n+1}^k \\ \vdots & \vdots & \vdots & \vdots \\ g_{m+1,1}^k & g_{m+1,2}^k & \cdots & g_{m+1,n+1}^k \end{bmatrix}, \mathbf{H}^k = \begin{bmatrix} h_{11}^k & h_{12}^k & \cdots & h_{1,n+1}^k \\ h_{21}^k & h_{22}^k & \cdots & h_{2,n+1}^k \\ \vdots & \vdots & \vdots & \vdots \\ h_{m+1,1}^k & h_{m+1,2}^k & \cdots & h_{m+1,n+1}^k \end{bmatrix}. \tag{9}$$

Like what we did in Eq. (3), we can transfer the two matrices into two vectors $\vec{g}^k$ and $\vec{h}^k$ for a compact expression.

$$\left.\begin{aligned} M\vec{g}^k = \vec{0} \\ a_{ij} - (a_{ij} - lb_{ij})x_{ij} \leq g_{ij}^k \leq a_{ij} + (ub_{ij} - a_{ij})x_{ij} \end{aligned}\right\} \tag{10}$$

$$\left.\begin{aligned} M\vec{h}^k = \vec{0} \\ a_{ij} - (a_{ij} - lb_{ij})x_{ij} \leq h_i^k \leq a_{ij} + (ub_{ij} - a_{ij})x_{ij} \end{aligned}\right\} \tag{11}$$

$$g_{i_k}^k \geq a_{i_k} + UPL_k \tag{12}$$

$$h_{i_k}^k \leq a_{i_k} - LPL_k \tag{13}$$

$\vec{g}^k$ and $\vec{h}^k$ are both feasible values for the suppressed table; thus, their satisfaction of Eq. (8) leads to constrain 10 and 11. Besides, the inequality in constraint 10 and 11 is equivalent to

$$\left.\begin{aligned} g_{ij}^k = a_{ij}, \text{ if } x_{ij} = 0 \\ lb_{ij} \leq g_{ij}^k \leq ub_{ij}, \text{ if } x_{ij} = 1 \end{aligned}\right\} \text{ and } \left.\begin{aligned} h_{ij}^k = a_{ij}, \text{ if } x_{ij} = 0 \\ lb_{ij} \leq h_{ij}^k \leq ub_{ij}, \text{ if } x_{ij} = 1 \end{aligned}\right\}$$

In summary, the cell suppression problem can be modeled as a mixed integer linear programming

(MILP) problem in Eq. (4), (5), (6), and for each primary cell $(i_k, j_k) \in \mathcal{P}$, Eq. (10), (11), (12), (13) with the binary variables $\vec{x}$ and the continuous variables $\vec{g}^k$ and $\vec{h}^k$. This MILP formulation was first proposed in Kelly et al. (1992) and in this paper, we adopted the compact version proposed in Fischetti and González (2000).

The mixed integer linear programming (MILP) model for cell suppression is proved to be strongly NP-hard in Kelly et al. (1992), suggesting that the existence of an efficient (i.e., polynomial time) algorithm for the exact solution for all possible instances is highly unlikely. Besides, Fischetti and Salazar (1999) and Fischetti and González (2000) pointed out that there are exponentially many constraints enforcing the protection level requirements. Restricted to computer performance and the algorithm, researchers like Kelly et al. (1992) and Cox (1995) can only propose some heuristic solutions to the large-scale MILP problem in the early stage. Later with the development of integer programming algorithm (such as branch-and-cut algorithm proposed in Padberg and Rinaldi (1991)), some exact optimal solutions were proposed, for example, in Fischetti and Salazar (1999) and Fischetti and González (2000). We will discuss these exact optimal solutions in the next section.

In Kelly et al. (1992), the objective function in Eq. (4) is actually approximated by its linear relaxation in each iteration to improve the computational efficiency:

$$\min \sum_{i=1}^{(m+1)} \sum_{j=1}^{(n+1)} c_{ij}(y_{ij}^{+} + y_{ij}^{-}) \tag{14}$$

where $y_{ij}^{+}, y_{ij}^{-} \geq 0$ are continuous variables. Cell (i, j) is suppressed if either $y_{ij}^{+} > 0$ or $y_{ij}^{-} > 0$.

This is essentially a minimal-cost-flow (MCF) problem and there exist many efficient algorithms such as Edmonds and Karp (1972); Ervolina and McCormick (1993); Goldberg and Tarjan (1989, 1990); Hassin (1983); Klein (1967); Orlin (1997).This MCF approximation to the objective function with other costs was widely applied in practice, such as Jewett (1993); Robertson (1993).

Cox (1995) built up the network optimization problem using Eq. (4), (5), (6). However, it deals with the protection level constraints in a different way from Eq. (10), (11), (12), (13) by finding the alternating cycles.

Each cell/arc in the network corresponds to two indicators $x_i^{+}, x_i^{-} \in \{0, 1\}$, representing flow increase and flow decrease, respectively. Then the objective function is

$$\min \sum_{i=1}^{(m+1)} \sum_{j=1}^{(n+1)} c_{ij}(x_{ij}^{+} + x_{ij}^{-}) \tag{15}$$

In the solution to this MILP problem, either $x_{ij}^{+} = 1$ or $x_{ij}^{-} = 1$ indicates the suppression of the cell. Assume that (i_k, j_k) is the primary cell for the current iteration, an alternating cycle γ containing arc, which connects row i_k and column j_k, can be found. Let $q(\gamma) = min\{a_{ij} : (i,j) \in \gamma\}$. To protect the primary cell (i_k, j_k), any cells/arcs in γ are selected as complementary cells, that will make the attacker's estimate of $a_{i_k j_k}$ to be $[a_{i_k j_k} - q(\gamma), a_{i_k j_k} + q(\gamma)]$. If $q(\gamma) \geq UPL_k$ and $q(\gamma) \geq LPL_k$, then these complementary cells can successfully protect the primary. If not, another alternating cycle γ containing arc (i_k, j_k) can be found to protect the remaining protection level.

Table 3 shows an example of this procedure. Given the primary cell (row_1, col_1) with $a_{11} = 1000$ and protection level $UPL_k = LPL_k = 23$, the first alternating cycle was selected as $\gamma_1 = \{(row_1, col_1), (col_1, row_2), (row_2, col_2), (col_2, row_1)\}$. $q(\gamma_1) = min\{\, a_{11} = 1000, a_{21} = 12, a_{22} = 10, a_{12} = 13 \,\} = 10$. Although the required protection level is 23, then the remaining protection requirement is $23 - 10 = 13$. Then the second alternating cycle was selected as $\gamma_2 = \{(row_1, col_1), (col_1, row_3), (row_3, col_3), (col_3, row_1)\}$. $q(\gamma_2) = min\{a_{11} = 1000, a_{31} = 17, a_{33} = 15, a_{13} = 18\,\} = 15 > 13$. The suppression of the complementary cells in γ_1 and γ_2 can successfully protect the primary cell (row_1, col_1) with required protection level 23.

The selection of the alternating cycle γ is achieved by solving the MILP problem. Different solutions are obtained by specifying different costs c_{ij}'s. For example, to minimize the total value of complementary cells for the current iteration primary cell (i_k, j_k), the costs are defined as

$$c_{ij} = \begin{cases} -1 - \sum_{(i,j) \neq (i_k, j_k)} c_{ij}, \text{ for } (i,j) = (i_k, j_k) & (a) \\ 1, \text{ for } (i,j) \in S & (b) \\ |S| + a_{ij}, \text{ otherwise,} & (c) \end{cases}$$

where S is the set of *previously suppressed* cells, i.e., previously handled primary cells and related complementary cells while excluding the current primary cell (i_k, j_k); $|S|$ is the number of previously suppressed cells. The large negative cost in (a) for the primary cell will force $x_{i_k j_k}^{+} = 1$ or $x_{i_k j_k}^{-} = 1$ for suppression; the cost of 1 in (b) encourages the selection of previously suppressed cells to protect the current

primary cell and therefore reduces the number/value of newly suppressed cell; the cost of $|S|+ a_{ij}$ in (c) for all other cells encourages small value cells to be selected in the solution. This is a pure integer programming problem, i.e., no continuous variable. However, in applications, a similar linear relaxation (MCF) to Kelly et al. (1992) is used for a faster solution.

The "heuristic" that the cost $c_{ij} = a_{ij}$, which represents the total value of suppressed cells exactly, manifests itself in Eq. (15). Further, this alternating cycle method can only provide symmetric protection which will lead to over-suppression for asymmetric protection requirements. Finally, for both heuristic solutions, the iterative method inevitably leads to over-suppression compared with the method of considering all primary cells together.

4.2 Exact Optimal Solutions to the Cell Suppression Problem

Fischetti and Salazar (1999) adopted the formulation of Eqs. (4), (5) and (6) but built up a new integer programming approach (without continuous variables $\vec{g}^k$ and $\vec{h}^k$) using the max-flow min-cut theorem, which will be discussed below.

The attacker's perspective can be used to derive the protection level constraints. Given a suppressed table with some primary cells $\mathcal{P}$ and complementary cells $\mathcal{C}$, suppose $SUP = \mathcal{P} \cup \mathcal{C}$. Fischetti and Salazar (1999) developed an incremental network $D(SUP)$ from the original network $D = (V, A)$ (See Fig. 2):

1. Remove all the arcs $(i, j) \in A \backslash SUP$.

2. Replace each arc $(i, j) \in SUP$ by two arcs, namely:
   – a forward arc (i, j) with capacity $q_{ij} = a_{ij} - lb_{ij}$
   – a reverse arc (i, j) with capacity $q_{ji} = ub_{ij} - a_{ij}$.

Each forward arc of $D(SUP)$ corresponds to a flow increase y^+_{ij} of an arc of D while each reverse arc corresponds to a decrease of the flow value y^-_{ij}. Then the feasible value for cell (i, j) is $f_{ij} = a_{ij} + y^+_{ij} - y^-_{ij}$. Recall that $[lb_{ij}, ub_{ij}]$ is the feasible range of the cell value a_{ij} which is known to the attacker, usually $lb_{ij} = 0$ and $ub_{ij} = +\infty$ for positive tables. Thus, $lb_{ij} \leq f_{ij} = a_{ij} + y^+_{ij} - y^-_{ij} \leq ub_{ij}$ leads to the capacity constraints $0 \leq y^+_{ij} \leq q_{ij} = a_{ij} - lb_{ij}$, $0 \leq y^-_{ij} \leq q_{ji} = ub_{ij} - a_{ij}$.

Definition 5. (Cut) *A cut is a node partition (S, T) such that s is in S and t is in T, where s is the source node and t is the sink node.*

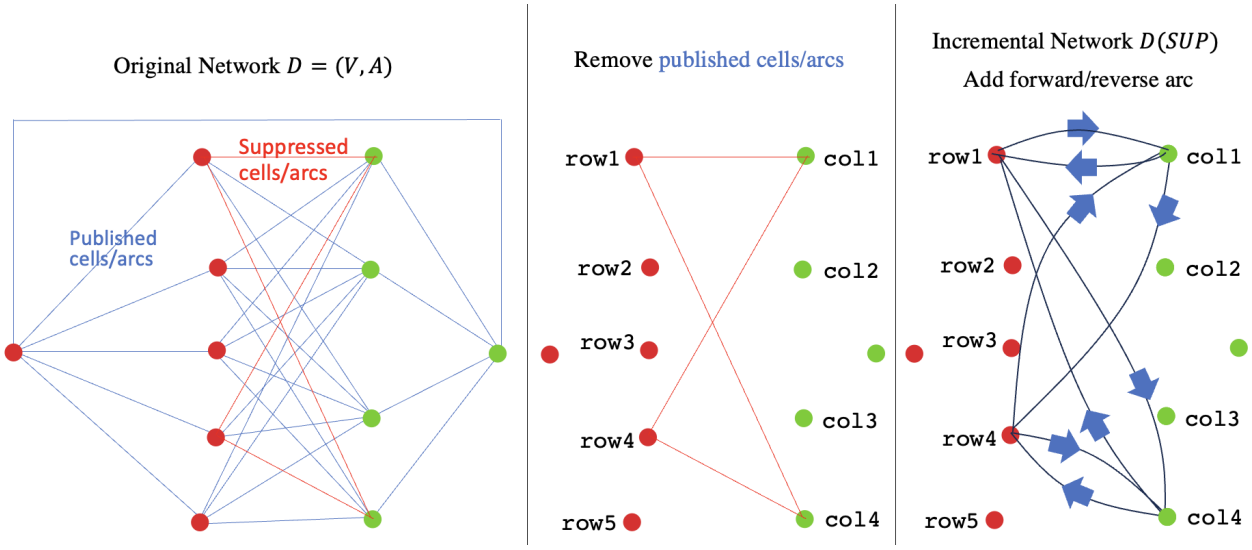

Figure 2: Procedure to generate the incremental network $D(SUP)$ from the original network $D = (V, A)$.

Definition 6. (Capacity of a cut) *Capacity* (S, T) = *sum of capacities of arcs leaving* S.

Theorem 1. (Max-Flow Min-Cut Theorem.) *In a flow network, the maximum amount of flow passing from the source to the sink is equal to the total capacity of the edges in a minimum cut, i.e., the smallest total capacity of the edges which if removed would disconnect the source from the sink.*

The interval estimate of a primary cell $a_{i_k j_k}$ from the attacker would be $[a_{i_k j_k} - max(y^-_{i_k j_k}), a_{i_k j_k} + max(y^+_{i_k j_k})]$. From the network flow theory, we have

$max(y^+_{i_k j_k})$	$max(y^-_{i_k j_k})$
= Max increase on cell (i_k, j_k)	= Max decrease on cell (i_k, j_k)
= Max-flow from $s = j_k$ to $t = i_k$	= Max-flow from $s = i_k$ to $t = j_k$
= Min-cut from $s = j_k$ to $t = i_k$	= Min-cut from $s = i_k$ to $t = j_k$
= the smallest total capacity q of the edges which which if removed would disconnect $s = j_k$ to $t = i_k$	= the smallest total capacity q of the edges which if removed would disconnect $s = i_k$ to $t = j_k$

,

To protect the primary cell (i_k, j_k), the attacker's estimate must be wider than the protection levels.

$$\begin{aligned}
& [a_{i_k j_k} - max(y^-_{i_k j_k}), a_{i_k j_k} + max(y^+_{i_k j_k})] \subseteq [a_{i_k j_k} - LPL_k, a_{i_k j_k} + UPL_k)] \\
\Leftrightarrow \quad & max(y^-_{i_k j_k}) \geq LPL_k, max(y^+_{i_k j_k}) \geq UPL_k \\
\Leftrightarrow \quad & \text{min } (j_k, i_k)\text{-cut } \geq LPL_k, \text{ min } (i_k, j_k)\text{-cut } \geq UPL_k \\
\Leftrightarrow \quad & \text{any } (j_k, i_k)\text{-cut } \geq LPL_k, \text{ any } (i_k, j_k)\text{-cut } \geq UPL_k
\end{aligned} \tag{16}$$

$$\Leftrightarrow \begin{array}{ll}
\displaystyle\sum_{(u,v)\in\delta^+(S)\setminus\{(j_k,i_k)\}} q_{uv}x_{uv} \geq UPL_k, & \text{for all } S \subset V : j_k \in S, i_k \notin S, \\
\displaystyle\sum_{(u,v)\in\delta^+(S)\setminus\{(i_k,j_k)\}} q_{uv}x_{uv} \geq LPL_k, & \text{for all } S \subset V : i_k \in S, j_k \notin S,
\end{array} \tag{17}$$

where $\delta^+(S)$ denotes the cut containing the arcs of $D(A)$ leaving a given $S \subseteq V$, and arc capacities q_{uv} previously defined. $D(A)$ is constructed as $D(SUP)$ with each arc being replaced by a forward arc and a reverse arc. The condition for all $S \subset V : j_k \in S, i_k \notin S$ implies that there are exponentially many protection level constraints for each primary cell.

A branch-and-cut algorithm for the exact optimal solution of the pure integer programming problem is proposed in Fischetti and Salazar (1999). Some tricks are used to reduce the number of constraints to improve the computational efficiency as well.

4.3 The Extension of the Linear Programming Formulation of Cell Suppression to High dimensional, Hierarchical and Linked Tables

Fischetti and González (2000) studied MILP formulation for the cell suppression problem proposed in Kelly et al. (1992). However, the protection level constraints are remodeled by employing the dual-block structure of the linear constraints for each primary cell $(i_k, j_k) \in \mathcal{P}$ – Eqs. (10), (11) and (12), (13) – using Bender's decomposition (Wolsey and Nemhauser (1999)). The idea is to use standard LP duality theory to project the auxiliary variables $\vec{g}^k$ and $\vec{h}^k$, $k = 1, ..., p$ away from the model and formulate a integer programming problem without any continuous variables. An enumerative algorithm in the space of the decision variable $\vec{x}$ was outlined for its exact optimal solution. Some near-optimal solutions can be also obtained from a linear relaxation of the integer programming problem. The significant advantage of this formulation is that it can be easily generalized to three or higher-dimensional tables, hierarchical tables, and linked tables.

Steel et al. (2013) used the framework of the MILP problem and extended it to high dimensional, hierarchical, and linked tables. The details on the linear constraints introduced by the complex table structure such as high dimensional, hierarchical, and linked were provided. Similar tricks to Fischetti and González (2000) were used to reduce the problem scale. A near-optimal solution was obtained using the Linear-programming Solver Cplex developed by IBM (See Cplex (2009)). A quickaudit procedure was applied to check the risk of under-suppression.

5. Summary

In this paper, five papers on classical linear programming formulation of the cell suppression problem for two-dimensional tables based on the network flow theory and its extension to high dimensional, hierarchical, and linked tables were reviewed. The development of the solutions from heuristic to optimal evolves along with the development of the (mixed) integer programming algorithm. Kelly et al. (1992) and Cox (1995) are among the pioneers in studying the cell suppression problem. The heuristic solutions were implemented for economic surveys and the Economic Census in 1990s by the U.S. Census Bureau and the Census of Agriculture conducted by Statistics Canada before 2021 (See Cox (1992), Robertson (1993) and Jewett (1993)).

Fischetti and Salazar (1999) and Fischetti and González (2000) contributed greatly to a theoretically sound and computationally efficient exact optimal solution to the cell suppression problem with an integer programming formulation. This work produces the software program τ-ARGUS designed to protect statistical tables with cell suppression that are currently used by statistical agencies in the European Union. The improvement on the cell suppression methodology proposed in Steel et al. (2013) was then applied to the surveys by the U.S. Census Bureau.

The motivation of this review paper is NASS's ongoing effort to upgrade the current cell suppression disclosure system that was developed in the 1990s based on primarily Cox (1995). The development of modern cell suppression frameworks has led NASS to develop a new disclosure system based on the efficient (mixed) integer linear programming algorithms, such as branch-and-cut, quantify the over-suppression of near-optimal solutions to the exact optimal solutions, and compare the performance on several commercial Linear-programming algorithms, such as Cplex (2009) and Gurobi Optimization, LLC (2023).

References

Cox, L. (1992). Solving confidentiality protection problems in tabulations using network optimization: a model for cell suppression in us economic censuses. In *Preproceedings of the International Seminar on Statistical Confidentiality, ISI-Eurostat, Dublin*.

Cox, L. H. (1995). Network models for complementary cell suppression. *Journal of the American Statistical Association*, 90(432):1453–1462.

Cplex, I. I. (2009). V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157.

Edmonds, J. and Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264.

Ervolina, T. R. and McCormick, S. T. (1993). Two strongly polynomial cut cancelling algorithms for minimum cost network flow. *Discrete Applied Mathematics*, 46(2):133–165.

Fischetti, M. and González, J. J. S. (2000). Models and algorithms for optimizing cell suppression in tabular data with linear constraints. *Journal of the American Statistical Association*, 95(451):916–928.

Fischetti, M. and Salazar, J. J. (1999). Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control. *Mathematical Programming*, 84(2).

Goldberg, A. V. and Tarjan, R. E. (1989). Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM (JACM)*, 36(4):873–886.

Goldberg, A. V. and Tarjan, R. E. (1990). Finding minimum-cost circulations by successive approximation. *Mathematics of Operations Research*, 15(3):430–466.

Gurobi Optimization, LLC (2023). Gurobi Optimizer Reference Manual.

Hassin, R. (1983). The minimum cost flow problem: a unifying approach to dual algorithms and a new tree-search algorithm. *Mathematical Programming*, 25:228–239.

Jewett, R. (1993). Disclosure analysis for the 1992 economic census. *Manuscript, Economic Programming Division, Bureau of the Census*.

Kelly, J. P., Golden, B. L., and Assad, A. A. (1992). Cell suppression: Disclosure protection for sensitive tabular data. *Networks*, 22(4):397–417.

Klein, M. (1967). A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science*, 14(3):205–220.

Orlin, J. B. (1997). A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78:109–129.

Padberg, M. and Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100.

Robertson, D. (1993). Cell suppression at statistics canada. In *Proceedings of the Annual Research Conference, US Bureau of the Census*, pages 107–131.

Steel, P., Fagan, J., Massell, P., Moore, J. r., Slanta, J., and Wang, B. (2013). Re-development of the cell suppression methodology at the us census bureau.

Wolsey, L. A. and Nemhauser, G. L. (1999). *Integer and combinatorial optimization*, volume 55. John Wiley & Sons.