The Computational Complexity of Some Rounding and Survey Overlap Problems

Kirk Pruhs, Computer Science Department
University of Pittsburgh, Pittsburgh, PA 15260

KEY WORDS: NP-complete, Zero-restricted Rounding, Unbiased Rounding, Controlled Rounding

1. Introduction

In this paper we examine the computational complexity of two classes of problems. The first class of problems involves rounding entries in multi-way tables subject to certain restrictions. The second class of problems involves maximizing the overlap between several surveys, with stratified design, on a common population. In particular, we investigate whether efficient algorithms exist for these problems. For some of these problems we give efficient algorithms. For the other problems, we use the theory of NP-completeness to show that it is highly unlikely that efficient algorithms exist for these problems.

The rounding problems that we consider are the zero-restricted rounding problem and the unbiased rounding problem. Given a table with rational entries, the goal in the zero-restricted rounding problem is to replace the entries in the table with adjacent integers in such a way that the marginals are maintained (this is called a zero-restricted rounding). Given a table with rational entries, the goal in the unbiased rounding problem is randomly generate a zero-restricted rounding in such a way that the expected value of the rounding of each entry is equal to the value of that entry. For applications of these rounding problems see [CFGH, Cox, HRF, IK]. Cox [Cox], and Causey, Cox, and Ernst [CCE] have given efficient algorithms for generating unbiased roundings of 2-way tables. Their algorithms run in time $O(n^3)$ on n by n tables. An algorithm A runs in time $O(f(n))$, for some function f, if for every sufficiently large input, A finishes after at most $c \cdot f(n)$ steps, where c is some constant and n is the size of the input. An unbiased rounding of a 1-way table of size n can be generated in time $O(n)$.

Causey, Cox, and Ernst [CCE], Hess and Srikantan [HS], Waterton [Wat], and Cox [Cox] posed the problem of how to generalize these results to 3-way tables. The first difficulty encountered when generalizing these results is that not all 3-way tables have zero-restricted roundings [CCE, HS]. Given a 3-way table T, one might still hope for an efficient algorithm that determines whether T has a zero-restricted rounding (or an unbiased rounding), and if so, generates such a rounding. Attempts to design efficient algorithms for this problem have been unsuccessful [Cox, HS, Wat]. In §3, we explain why these attempts were unsuccessful by showing that both the problem of determining whether a 3-way table has a zero-restricted rounding, and the problem of determining whether a 3-way table has an unbiased rounding, are NP-hard. We discuss, in §2, the implications of a problem being NP-hard.

An instance of a survey overlap problem consists of several survey sampling problems, with stratified design, on a common population. (We define stratification in §2.) It is possible that both the stratification and the selection probabilities of the sampling units are different in each survey. In survey overlap problems we make the assumption that the cost of sampling is roughly proportional to the total number of units sampled in all of the surveys, i.e., it is cheaper to sample the same unit twice than it is to sample two distinct units. Minimizing the number of distinct units chosen in the different surveys would then minimize the cost. Hence, the goal in the survey overlap problem is to solve each survey in such a way as to minimize the maximum number of units in the union of the samples.

In section 4, we give an $O(n^2)$ time algorithm for instances of the survey overlap problem that consist of two singularly stratified survey sampling problems. This algorithm is optimal, in terms of minimizing the total number of elements sampled, both in the expected case and in the worst case. We then show that if we generalize this problem by allowing three singularly stratified surveys, or two doubly stratified surveys, then the survey overlap problem becomes NP-hard. We also show that the survey overlap problem is NP-hard for instances consisting of an arbitrary number of unstratified surveys.

2. Definitions

We assume familiarity with standard definitions and concepts from graph theory; see for example. An *edge 3-coloring* of a multigraph is an assignment of one of three colors to each edge that has the property that each pair of edges incident on a common vertex are assigned distinct colors. A *vertex 3-coloring* of a multigraph G is an assignment of one of three colors to each vertex of G,

with the property that each pair of adjacent vertices are assigned different colors. For notational convenience we will represent the three colors by the integers 1, 2, and 3.

An m-way *array*, $A = A_1 \times A_2 \times \ldots A_m$, is the Cartesian product of m finite sets. Each A_i is called a *stratum*. An m-way *table* T is an assignment of a positive number, $T(a_1, a_2, \ldots, a_m)$, to each element $(a_1, a_2, \ldots, a_m)$ of A. Each a_i is called an *index*, and $T(a_1, a_2, \ldots, a_m)$ is called an *entry* of T.

We denote the floor and ceiling of a number x by, $\lfloor x \rfloor$ and $\lceil x \rceil$, respectively. An integer k is a *rounding* of rational number x with respect to a *rounding base* B if $\frac{k}{B} = \lfloor \frac{x}{B} \rfloor$, or $\frac{k}{B} = \lceil \frac{x}{B} \rceil$. A table R is a *rounding* of T with respect to a rounding base B, if R and T have the same underlying array, and each entry of R is a rounding with respect to B of the corresponding entry in T. From now on we assume, without loss of generality, that $B = 1$, and that the range of T is $[0, 1)$ [CCE]. A k-way *hyperplane*, $0 \leq k \leq m$, of A is obtained fixing $m - k$ of the indices, and is denoted in the following manner. $A(a_1, *, *, a_4, *)$ is the 3-way hyperplane derived from A by considering only those elements of A whose the first index is a_1 and whose fourth index is a_4. We call a 1-way hyperplane a *line*, and a 2-way hyperplane a *plane*. For a table T and a hyperplane H, we define $\#T(H)$ to be $\sum_{a \in H} T(a)$. A rounding R of T *preserves* a hyperplane H if $\#R(H)$ is a rounding of $\#T(H)$.

R is a *zero-restricted* rounding of T if R preserves all hyperplanes. A probabilistic procedure generates a *fair* rounding R of a table T if the expected value of each $R(a_1, a_2, \ldots, a_m)$ is $T(a_1, a_2, \ldots, a_m)$. A probabilistic procedure generates an *unbiased* rounding R of a table T if the procedure fairly generates a zero-restricted rounding T.

In the m-way stratified sampling problem the population is stratified by m variables that are hopefully correlated with the measure variable. To reduce the variance of the measure variable a sample whose distribution among the strata mirrors the populations distribution as closely as possible. We can formally define the m-way stratified sampling problem in a manner similar to the way that we defined unbiased rounding, the only difference being that each strata may have more than one sampling unit.

We now explain the tools that we use to provide evidence that some problems are computationally difficult. The class P is defined as the class of problems solvable in time $O(p(n))$, for some polynomial $p(n)$. The class P is generally regarded as the class of problems that can be solved by computationally feasible algorithms [GJ]. The class NP is defined as the the class of problems for which it is possible to, in polynomial time, guess at a potential solution and then verify the correctness of that potential solution. As an example of a problem in NP, consider the edge 3-coloring problem. Given a trivalent multigraph G, the *edge 3-coloring problem* is the problem of determining whether G has an edge 3-coloring. It easy to, in polynomial time, guess a color for each edge, and then verify that each vertex has exactly one edge incident to it of each color.

There are many problems, such as the edge 3-coloring problem, that are known to be in NP, but are not known to be P. Garey and Johnson's book [GJ] contains a 100 page list of such problems arising from such diverse areas as graph theory, network design, algebra, number theory, math programming, and logic. The $P = ?NP$ problem is the problem of determining whether all problems in NP are solvable by polynomial-time algorithms. The $P = ?NP$ problem is the most important open problem in theoretical computer science, and has been cited as one of the ten most famous open problems in mathematics.

An important concept in the study of the $P = ?NP$ problem is NP-completeness (or NP-equivalence). A problem $\mathcal{P}$ is *NP-hard* if a polynomial-time algorithm for $\mathcal{P}$ implies that every problem in NP has a polynomial-time algorithm. A problem is *NP-equivalent* if it is in NP and it is NP-hard. Most natural problems, that are known to be in NP, but are not known to be in P, are NP-equivalent.

One way to show that a problem $\mathcal{P}$ is NP-hard is to exhibit a polynomial-time reduction from some known NP-hard problem $\mathcal{C}$ to $\mathcal{P}$. A function r from instances of $\mathcal{C}$ to instances of $\mathcal{P}$ is a *polynomial-time reduction* if:

1. $r(x)$ can be computed in time polynomial in the size of x.
2. Each instance x of $\mathcal{C}$ has a solution if and only if the instance $r(x)$ of $\mathcal{P}$ has a solution.

If $\mathcal{C}$ is polynomial-time reducible to $\mathcal{P}$, and there is a polynomial-time algorithm for deciding whether an instance of $\mathcal{P}$ has a solution, then there is a polynomial-time algorithm for deciding whether an instance of $\mathcal{C}$ has a solution. Holyer [Hol] showed that the edge 3-coloring problem is NP-hard.

It is (almost) unanimously believed that $P \neq NP$. If this is true, then by definition no NP-hard

problem can have a polynomial-time algorithm. At the very least the fact that a problem $\mathcal{P}$ is NP-hard implies that one should expect the task of finding a polynomial-time algorithm for $\mathcal{P}$ to be extremely difficult.

The following variant of edge coloring is NP-hard [Pru]:

Problem 2.1:
INSTANCE: A trivalent multigraph G and a vertex 3-coloring f of G.
QUESTION: Does G have an edge 3-coloring?

3. Rounding

To show that the unbiased rounding problem and the zero-restricted rounding problems are NP-hard for 3-way tables we reduce from problem 2.1.

Reduction 3.1: Given a trivalent multigraph $G = (V, E)$, and a vertex 3-coloring f of G, we construct a 3-way table T from G as follows. Let stratum A_j, $j = 1, 2, 3$, be defined by:

$$A_j = \{v_i | v \in V, f(v) = j, \text{ and } i = 1, 2, 3\} \cup$$

$$\{(v, w) \in E | f(v) \neq j \text{ and } f(w) \neq j\}$$

Stratum A_j consists of three copies of each vertex colored j, and one copy of each edge that has no vertex incident to it colored j. We define $[v_i, w_i, e]$, for $e = (v, w)$ and $i = 1, 2, 3$, to be the unique element of A with indices v_i, w_i and e. More precisely:

$$[v_i, w_i, e] = \begin{cases} (v_i, w_i, e) & \text{if } f(v) = 1 \ \& \ f(w) = 2; \\ (w_i, v_i, e) & \text{if } f(v) = 2 \ \& \ f(w) = 1; \\ (e, v_i, w_i) & \text{if } f(v) = 2 \ \& \ f(w) = 3; \\ (e, w_i, v_i) & \text{if } f(v) = 3 \ \& \ f(w) = 2; \\ (v_i, e, w_i) & \text{if } f(v) = 1 \ \& \ f(w) = 3; \\ (w_i, e, v_i) & \text{if } f(v) = 3 \ \& \ f(w) = 1. \end{cases}$$

For each edge $e = (v, w)$, and each $i = 1, 2, 3$, let $T[v_i, w_i, e] = \frac{1}{3}$ (rounding this entry to 1 is equivalent to coloring the edge e the color i). Let all other entries in T be 0. ∎

Lemma 3.2: Let T be a table constructed by reduction 3.1 from a multigraph G and a vertex 3-coloring f. Then T has a zero-restricted rounding if and only if G has an edge 3-coloring. Furthermore, T has a zero-restricted rounding if and only if T has a unbiased rounding.

Theorem 3.3: Given a 3-way table T, the problem of determining whether T has a zero-restricted rounding is NP-hard.

Theorem 3.4: Given a 3-way table T, the problem of determining whether T has an unbiased rounding is NP-hard.

We next show that various relaxations of these rounding problems remain NP-hard. One obvious relaxation, that would still be useful in some applications, is to require that only planes, and not lines, be preserved by the rounding procedure. Unfortunately, lemma 3.5 implies that, in the worst case, this relaxation does not make the rounding problems any easier.

Lemma 3.5: Let T be a table constructed by reduction 3.1. If a rounding R of T preserves planes, then R preserves lines.

Corollary 3.6: Given an 3-way table T, the problem of determining whether T has a rounding that preserves planes is NP-hard.

Corollary 3.7: Given an 3-way table T, the problem of determining whether T has a fair rounding T that preserves planes is NP-hard.

We now show that both zero-restricted rounding and unbiased rounding remain NP-hard for 3-way tables that have the property that one stratum is of size at most six. In other words, these rounding problems remain hard for "flat" tables. We reduce from the edge coloring problem.

Reduction 3.8: Given a trivalent multigraph $G = (V, E)$, we construct a 3-way table T from G as follows. The strata are $A_1 = \{e_j | e \in E, j = 1, 2\}$, $A_2 = \{v_i | v \in V, i = 1, 2, 3\}$, and $A_3 = \{1, \ldots, 6\}$. For each $e = (v, w) \in E$, and each $i = 1, 2, 3$, let $T(e_1, v_i, 1) = \frac{1}{3}$, and let $T(e_2, w_i, 1) = \frac{1}{3}$ (the pairing of v with e_1 was arbitrary, we could have just as easily paired w with e_1). Rounding an entry of the form $T(e_1, v_i, 1)$ or $T(e_2, w_i, 1)$, $i \in \{1, 2, 3\}$, to 1 is equivalent to coloring the edge e the color i. We then use the following construction to guarantee that entries of this form are rounded in the same direction.

Sequentially consider the edges. Assume we are considering the edge $e = (v, w)$. For each $i = 1, 2, 3$, find a plane of the form $A(*, *, k_i)$, $2 \leq k_i \leq 6$, for which none of the entries in the rows $A(*, v_i, k_i)$ and $A(*, w_i, k_i)$ have been defined yet. It is always possible to find such a k_i because each of the vertices v and w are adjacent to at most two other vertices. Let $T(e_1, v_i, k_i) = T(e_2, w_i, k_i) = \frac{2}{3}$, and let $T(e_1, w_i, k_i) = T(e_2, v_i, k_i) = \frac{1}{3}$.

After this construction has been completed for each edge, let the unassigned table entries be assigned 0. ∎

Lemma 3.9: Let T be a table constructed by reduction 3.8 from a multigraph G. Then T has a zero-restricted rounding if and only if G has an edge

3-coloring. Furthermore, T has a zero-restricted rounding if and only if T has a unbiased rounding.

Theorem 3.10: Given a 3-way table T, with the size of one stratum being at most six, the problem of determining whether T has a zero-restricted rounding is NP-hard.

Theorem 3.11: Given a 3-way table T, with the size of one stratum being at most six, the problem of determining whether T has a unbiased rounding is NP-hard.

We end this section by noting that all of the NP-hardness results in this section can easily be generalized to tables of dimension greater than three.

4. Survey Overlap

The *2-overlap problem* is a special case of the survey overlap problem in which the instance consists of two 1-way stratified sampling problem instances. We begin this section with an $O(n^2)$ algorithm for the 2-overlap problem. Our algorithm is motivated by Cox's [Cox] algorithm for generating unbiased roundings of 2-way tables.

We now define the 2-overlap problem more formally. Assume that the population is $X = \{x_1, \ldots, x_n\}$. Let the first 1-way stratified sampling problem instance consist of X, associated selection probabilities $\{p_1, \ldots, p_n\}$, and a partition $\mathcal{C}$ of X. Similarly, let the second 1-way stratified sampling problem instance consist of X, associated selection probabilities $\{q_1, \ldots, q_n\}$, and a partition $\mathcal{D}$ of X. The goal in the 2-overlap problem is to probabilistically generate two samples, S and T, from X in way such a way that the size of the maximum possible size of $S \cup T$ is minimized and the following criteria are satisfied:

Probability Constraints:

1. Each unit $x_i \in X$ is included in S with probability p_i.
2. Each unit $x_i \in X$ is included in T with probability q_i.

Structural Constraints:

1. The number of units in S is either the floor of, or the ceiling of, $u = \sum_{i=1}^{n} p_i$.
2. The number of units in T is either the floor of, or the ceiling of, $v = \sum_{i=1}^{n} q_i$.
3. The number of units in S from each $C \in \mathcal{C}$ is either the floor of, or the ceiling of, $\#C = \sum_{x_i \in C} p_i$.
4. The number of units in T from each $D \in \mathcal{D}$ is either the floor of, or the ceiling of, $\#D = \sum_{x_i \in D} q_i$.

We define the *overlap* between two samples to be the number of units common to both samples. In the 2-overlap problem, minimizing the total number of units sampled is equivalent maximizing the overlap. Each $x_i \in X$ can appear in both S and T with probability at most $min(p_i, q_i)$. Hence, an upper bound for the expected overlap is $d = \sum_{i=1}^{n} min(p_i, q_i)$. The expected overlap of our algorithm will be d, and the worst case overlap will be $\lfloor d \rfloor$. Hence, our algorithm guarantees optimal overlap, both in the average case, and in the worst case.

Samples S and T, which satisfy the structural constraints, can be viewed as 0/1 solutions to the following equations: (A unit x_i is included in S if $\tilde{p}_i = 1$, and in T if $\tilde{q}_i = 1$.)

(1) $\lfloor u \rfloor \leq \sum_{x_i \in X} \tilde{p}_i \leq \lceil u \rceil$.
(2) $\lfloor v \rfloor \leq \sum_{x_i \in X} \tilde{q}_i \leq \lceil v \rceil$.
(3) $\forall C \in \mathcal{C} \quad \lfloor \#C \rfloor \leq \sum_{x_i \in C} \tilde{p}_i \leq \lceil \#C \rceil$.
(4) $\forall D \in \mathcal{D} \quad \lfloor \#D \rfloor \leq \sum_{x_i \in D} \tilde{q}_i \leq \lceil \#D \rceil$.
(5) $\sum_{i=1}^{n} min(\tilde{p}_i, \tilde{q}_i) \geq d$.

We begin the algorithm by letting each $\tilde{p}_i = p_i$, and letting each $\tilde{q}_i = q_i$. The initial values of the $\tilde{p}_i$'s and the $\tilde{q}_i$'s then satisfy the above equations. The first phase of the algorithm transforms some of the $\tilde{p}_i$'s and $\tilde{q}_i$'s to 0/1 probabilities, in such a way that (1)-(5) remain valid. This transformation uses what we call a balanced adjustment. Let P and M be two disjoint subsets of the variables $\tilde{p}_1, \ldots, \tilde{p}_n$ and $\tilde{q}_1, \ldots, \tilde{q}_n$ that satisfy the following two conditions:

1) Each variable in $P \cup M$ is non 0/1.
2) Adding any amount to the value of each variable in P (M), and subtracting that same amount from the value of each variable in M (P), will not affect the veracity of (1)-(5).

Let $a^+ = min(1 - max(P), min(M))$, and $a^- = min(min(P), 1 - max(M))$. Given such sets, a *balanced adjustment* consists of randomly executing one of the following two assignments. With probability $\frac{a^-}{a^+ + a^-}$, add a^+ to the value of each variable in P, and subtract a^+ from the value of each variable in M. With probability $\frac{a^+}{a^+ + a^-}$, subtract a^- from the value of each variable in P, and add a^- to the value of each variable in M. A balanced adjustment causes at least one $\tilde{p}_i$, or one $\tilde{q}_i$, to become 0/1. The fairness constraints are not violated because the expected change of the value of each variable is zero.

Throughout the algorithm we maintain a vertex labeled bipartite multigraph G. The vertices of G are the classes in $\mathcal{C}$, and the classes in $\mathcal{D}$. The value of the $\tilde{p}_i$'s and the $\tilde{q}_i$'s determine the edges

and the labels. A unit x_i is *free* if exactly one of $\tilde{p}_i$ and $\tilde{q}_i$ is 0/1. A unit x_i is *bound* if neither of $\tilde{p}_i$ or $\tilde{q}_i$ is 0/1. For $x_i \in X$, with $x_i \in C \in \mathcal{C}$ and $x_i \in D \in \mathcal{D}$, edges and labels are added to G in the following manner:

1. If x_i is bound, then there is an undirected edge (C, D).
2. If x_i is free, with $\tilde{p}_i = 0$ (1), then a label of 0 (1) is added to D. In this case, the variable $\tilde{q}_i$ is called a 0-*mate* (1-*mate*) of D.
3. If x_i is free, with $\tilde{q}_i = 0$ (1), then a label of 0 (1) is added to C. In this case, the variable $\tilde{p}_i$ is called a 0-*mate* (1-*mate*) of C.

Each vertex may be labeled with any number of 0's and 1's, or may have no label. We will freely switch interpretations between edges and units, and between vertices and equivalence classes.

The goal of the first phase of the algorithm is to simplify G. Each simplification step consists of a balanced adjustment. We now define the three types of balanced adjustments used in the first phase.

Cycle Step: Let H be a cycle in G, with the edges in H alternately designated odd and even. Let P (M) be the set consisting of the variables $\tilde{p}_i$ and $\tilde{q}_i$, such that x_i is an odd (even) edge in H. A cycle step then consists of performing a balanced adjustment on these sets.

Pair Step: Let a and b be two 0-mates, or two 1-mates, of some vertex in G. Let $P = \{a\}$ and let $M = \{b\}$. A pair step then consists of performing a balanced adjustment on these sets.

Good Path Step: Let H be a simple path between two vertices y and z in G. Designate the edges alternately as odd and even, with an odd edge incident to y. H is then called a *good path* if it satisfies one following conditions:

1. H is of odd length, and satisfies one of the following conditions:
   a. y has a 1-mate a and z has a 0-mate b.
   b. y has a 1-mate a and z is a vertex of degree 1 with no label.
2. H is of even length, and satisfies one of the following conditions:
   a. y has a 0-mate a and z has a 0-mate b.
   b. y has a 1-mate a and z has a 1-mate b.
   c. y has a 0-mate a and z is a vertex of degree 1 with no label.
   d. Both y and z are vertices of degree 1 with no label.

Let H be a good path as defined above. If H satisfies more than one condition in the above definition, then pick a condition arbitrarily for the next steps. Let P (M) contain the variables $\tilde{p}_i$ and $\tilde{q}_i$, such that x_i that is an odd (even) edge in H. In case 1, also add the mate a, and the mate b, if it exists, to M. In case 2, add the mate a, if it exists, to M, and add the mate b, if it exists, to P. A good path step consists of performing a balanced adjustment on these sets.

The cycle step is applied repeatedly, updating G after each step, until G becomes acyclic. We now assume that G is acyclic. If y and z are vertices in the same connected component of G, we denote the unique path between them as $P(y, z)$. A simple path H is *maximal* if no other simple path properly contains H. In the following two theorems we give sufficient conditions for G to contain a good path. These theorems can be proved by exhaustively enumerating the possibilities.

Lemma 4.1: If H is a tree in G, with at least three leaves, x, y, and z, then one of $P(x, y)$, $P(x, z)$, or $P(y, z)$ is a good path.

Lemma 4.2: Let H be a maximal simple path between vertices x and z in G. If H contains an internal labeled vertex y then one of $P(x, y)$, $P(y, z)$, or $P(x, z)$ is a good path.

While G contains trees with at least three leaves, or maximal simple paths with labeled internal vertices, the algorithm repeatedly finds a good path, performs a good path step, and updates G. Next, the algorithm repeatedly finds vertices in G that have two 0-mates, or two 1-mates, performs a pair step, and updates G.

G is now of the following simple form. It is the disjoint union of simple paths and isolated vertices. Each vertex has at most one label. No internal vertex on a path can have a label. One consequence of this is that each equivalence class $C \in \mathcal{C}$ ($D \in \mathcal{D}$) contains at most two units x_i and x_j with $\tilde{p}_i$ and $\tilde{p}_j$ ($\tilde{q}_i$ and $\tilde{q}_j$) being non 0/1. At this point no further balanced adjustments may be possible, and phase 1 is finished. Phase 1 requires at most $O(n^2)$ time because each simplification step can be performed in linear time, and each simplification step makes at least one $\tilde{p}_i$, or one $\tilde{q}_i$, 0/1.

The second phase of the algorithm is divided into two parts. To begin the first part of phase two associate with each unit $x_i \in x$ a probability, $h_i = min(\tilde{p}_i, \tilde{q}_i)$. To guarantee maximum overlap we need to include each x_i in both samples with probability h_i. Note that if either $\tilde{p}_i$ or $\tilde{q}_i$ is 0, then $h_i = 0$ and x_i will not be included in both samples. Similarly, if both $\tilde{p}_i$ and $\tilde{q}_i$ are 1, then $h_i = 1$ and x_i will be included in both samples. We call units x_i and x_j in H a *C-pair* (*D-pair*) if they share a

common equivalence class in $\mathcal{C}$ ($\mathcal{D}$), and both $\tilde{p}_i$ and $\tilde{p}_j$ ($\tilde{q}_i$ and $\tilde{q}_j$) are non 0/1. A unit $x_i \in H$ is a *singleton* if it does not occur in a pair. We order H so that all pairs x_i and x_j are consecutive. This ordering is possible because of the simple form of G. We now use systematic sampling to select the units to be included in both samples.

Systematic Sampling: Let $g_i = \sum_{j<i} h_j$. Remember that $d = \sum_{i=1}^{n} h_i$. We conceptually associate with each x_i a half-open interval, $I_i = [g_i, g_{i+1})$. The algorithm first generates a random number q in the range [0,1). Let $Q = \{q + j | j = 0, \ldots, \lceil d \rceil - 1\}$. We select a unit x_i if I_i contains a point in Q.

In systematic sampling $\lfloor d \rfloor$ or $\lceil d \rceil$ units are selected because the sum of the lengths of the intervals is d. Each unit x_i is selected with probability h_i because the length of interval I_i is h_i. For each pair x_i and x_j, the ordering of H, and the fact that $h_i + h_j \leq 1$, guarantees that both x_i and x_j will not be selected. In the second part of phase two we finish the sample generation separately for each survey in linear time; we omit the details.

We next show that the most obvious ways to generalize the 2-overlap problem lead to NP-hard problems.

Theorem 4.3: Given an instance of the survey overlap problem $\mathcal{P}$, which consists of three 1-way stratified sampling problems, and an integer m, the problem of determining whether $\mathcal{P}$ has a solution in which no more than m units will be sampled is NP-hard.

Theorem 4.4: Given an instance of the survey overlap problem $\mathcal{P}$, which consists of one 1-way stratified sampling problem and one 2-way stratified sampling problem, and an integer m, the problem of determining whether $\mathcal{P}$ has a solution in which no more than m units will be sampled is NP-hard.

Theorem 4.5: Given an instance of the survey overlap problem $\mathcal{P}$, which consists of an arbitrary number of 0-way stratified sampling problems, and an integer m, the problem of determining whether $\mathcal{P}$ has a solution in which no more than m units will be sampled is NP-hard.

5. Conclusion

Several problems in this paper fall into a class of problems called controlled selection problems. These problems have the property that not all samples are acceptable. See [PM] for a further discussion of the computational complexity of these problems.

We are currently working on finding algorithms for zero-restricted rounding that are efficient on "average", ie. efficient on almost all tables. The computational complexity of the problem of generating a controlled rounding [Cox] of tables with dimension greater than two remains an intriguing open problem.

References

[CCE] B. D. Causey, L.H. Cox, and L.R. Ernst, Applications of Transportation Theory to Statistical Problems, *J. of the American Statistical Association*, **80**, 1985, pp. 903-909.

[CFGH] L. Cox, J. Fagan, B. Greenberg, and R. Hemming, Research at the Census Bureau into Disclosure Avoidance Techniques for Tabular Data, *Proceedings of the Survey Research Section, American Statistical Association*, 1986, pp. 135-160.

[Cox] L. H. Cox, A Constructive Procedure for Unbiased Controlled Rounding, *J. of the American Statistical Association*, **82**, 1988, pp. 520-524.

[GJ] M. Garey and D. Johnson, *Computers and Intractability, A Guide to NP-completeness*, W.H. Freeman and Company, New York, 1979.

[Hol] I. Holyer, The NP-completeness of Edge Coloring, *SIAM J. Comput.*, **10**, 1981, pp.718-720.

[HRF] I. Hess, T. Fitzpatrick and D. Riedel, *Probability Sampling of Hospitals and Patients*, 2nd edition, Health Administration Press, Ann Arbor, Michigan, 1975.

[HS] I. Hess and K.S. Srikantan, Some Aspects of the Probability Sampling Technique of Controlled Selection, *Health Services Research*, Summer, 1966, pp. 8-52.

[IK] K. Ireland, and S. Kullback, Contingency Tables with Given Marginals, *Biometrika* **55**, 1968, pp. 179-188.

[PM] K. Pruhs and U. Manber, The Complexity of Controlled Selection, *16th International Colloquium of Automata, Languages, and Programming*, Stressa, Italy, July 1989.

[Wat] J. Waterton, An Exercise in Controlled Selection, *Applied Statistics*, **32**, 1983, 150-164.