RESEARCH REPORT SERIES
(*Computing #2002-02)*

An Efficient Formulation of Age Comparison in the DISCRETE Edit System

Bor-Chung Chen and William E. Winkler

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

Report Issued: June 10, 2002

Disclaimer: This paper reports the results of research and analysis undertaken by Census Bureau staff. It has undergone a Census Bureau review more limited in scope than that given to official Census Bureau publications. This paper is released to inform interested parties of ongoing research and to encourage discussion of work in progress.

An Efficient Formulation of Age Comparisons in the DISCRETE Edit System[*]

Bor-Chung Chen and **William E. Winkler**

Abstract

The DISCRETE edit system, based on the Fellegi and Holt model [1976] of editing, contains two major components: edit generation and error localization. The set covering problem (SCP) is formulated with constraint matrices many times in both components. Therefore, an efficient set covering algorithm is critical to the overall performance of the DISCRETE edit system. The design of a set covering algorithm (Chen [1998]) provides a major performance improvement for the DISCRETE edit system. The size of the constraint matrices is a very important factor to have an efficient set covering algorithm. The age comparison approach described in Chen, Winkler, and Hemmig [2000] creates a huge number of new variables and edit rules in the set covering algorithm of the edit generation and error localization of the DISCRETE edit system. The dimension of the constraint matrices created is an increasing function of the number of variables and the number of edit rules. In this paper, we will describe an efficient formulation and simple implementation of the age comparisons in surveys that have the age field.

KEY WORDS: Explicit Edits, Redundant Covers, Subcovers, Integer Programming, Optimization

1 Introduction

The DISCRETE edit system (Winkler and Petkunas [1996]) is designed for general edits of discrete data. It utilizes the Fellegi-Holt model of editing and contains two major components: edit generation and error localization. An edit-generation algorithm, called the EGE algorithm, for the DISCRETE edit system was described in Winkler [1997]. The EGE algorithm is a much faster alternative to Algorithm 1, called the GKL algorithm, of Garfinkel, Kunnathur, and Liepins [1986]. In both of the EGE and GKL algorithms, the set covering routine is invoked many times to generate new implicit edits. Therefore, an efficient algorithm for the set covering problem becomes highly desirable to reduce the computation time of the edit generation. In error localization, the set covering problem, which, in fact, is an integer linear programming problem, is used to identify the minimum number of fields in an erroneous record to change in order to pass all the edits.

[*] This paper reports the results of research and analysis undertaken by Census Bureau staff. It has undergone a more limited review than official Census Bureau publications. This report is released to inform interested parties of research and to encourage discussion.

The SCP in the DISCRETE edit system is applied twice, one in edit generation and the other in error localization. The first application in error localization is to find the minimal set of fields (the optimal solution) of a failed record to be modified to satisfy all explicit and implicit edits. The SCP is invoked once for each failed record. The second application in edit generation is to find all the minimal sets of edits that are unioned to cover all possible values of a field, called a generating field. The second application is NOT to find an optimal solution but to find all prime cover solutions to the SCP. In either application, the size of the constraint matrices that form the SCP is an increasing function of the number of variables and the number of edit rules. The performance of the set covering algorithm is much better if the size of the constraint matrices is smaller. In Chen, Winkler, and Hemmig [2000], the age field is used to compare the ages between the household members; such as, the age of the householder must be at least 15 years older than the children. In this paper, a new formulation of the age comparisons is described to reduce the size of the constraint matrices in the set covering problem.

We will use the following notations in this paper: $\boldsymbol{a}= (a_1, a_2, \ldots, a_n)$ has n fields. $a_i \in A_i$ for each i, $1 \leq i \leq n$, where A_i is the set of possible values or coded values which may be recorded in Field i. $|A_i| = n_i$. If $a_i \in A_i^o \subset A_i$, we also say

$$\boldsymbol{a} \in \boldsymbol{A}_i^o = A_1 \times A_2 \times \ldots \times A_{i-1} \times A_i^o \times A_{i+1} \times \ldots \times A_n,$$

in which record $\boldsymbol{a}$ fails edit $\boldsymbol{A}_i^o$. The code space is $A_1 \times A_2 \times \ldots \times A_n = \boldsymbol{A}$. If edit $\boldsymbol{A}^o$ is a subset of edit $\boldsymbol{B}^o$ ($\boldsymbol{A}^o \subset \boldsymbol{B}^o$), then edit $\boldsymbol{A}^o$ is dominated by edit $\boldsymbol{B}^o$ and is therefore a redundant edit.

2 Background

The objective of error localization is to find the minimum number of fields to change if a record fails some of the edits. It can be formulated as a set covering problem. Let $\bar{E} = \{E^1, E^2, \cdots, E^m\}$ be a set of edits failed by a record $\boldsymbol{y}$ with n fields, consider the set covering problem:

$$\begin{array}{ll} \text{Minimize} & \sum_{j=1}^{n} c_j x_j \\ \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \geq 1, \quad i = 1, 2, \cdots, m \end{array} \tag{1}$$

$$x_j = \begin{cases} 1, & \text{if field } j \text{ is to be changed;} \\ 0, & \text{otherwise,} \end{cases}$$

where

$$a_{ij} = \begin{cases} 1, & \text{if field } j \text{ enters } E^i; \\ 0, & \text{otherwise,} \end{cases}$$

and c_j is a measure of "confidence" in field j. We need to get $\bar{E}$ from a *complete* set of edits to obtain a meaningful solution to (1). A complete set of edits is the set of explicit (initially specified) edits and all essentially new implied edits derived from them. The dimension of the constraint matrix (a_{ij}) of 0s and 1s associated with (1) is $m \times n$. The size of the preorder forest of the set covering algorithm described in Chen [1998] is $2^n - 1$. The preorder forest is a collection of tree data structures that provide a sequence, called *ranking*, of the n column vectors in the constraint matrix (a_{ij}) to be included in a possible cover solution to (1). The size of a preorder forest is the number of nodes in its collection of tree data structures and is therefore one of the important factors that affect the efficiency of a set covering algorithm.

If $\boldsymbol{x}$ is a prime cover solution to (1) and $K = \{r \mid x_r = 1\} \subset \{1, 2, \cdots, n\}$, then for each $k \in K$ we may change the value of field f_k to a value from

$$B_k^* = \overline{\bigcup_{j \in J} A_k^j} = \bigcap_{j \in J} \overline{A_k^j},$$

where $J = \{j \mid 1 \leq j \leq m,\ f_k \text{ is an entering field of } E^j\}$. The new imputed record $\boldsymbol{y}_1$, which has different value of $f_k\ \forall\ k \in K$ from the record $\boldsymbol{y}$, will pass all edits. Note that $B_k^* \neq \emptyset$. If B_k^* were a empty set, then $\bigcup_{j \in J} A_k^j$ would be equal to A_k and an essentially new implicit edit would have been generated and included in the set of $\bar{E}$.

To obtain a *complete* set of edits, implicit edits are needed. Implicit edits may be implied logically from the initially specified edits (or explicit edits). Implicit edits give information about explicit edits that do not originally fail but may fail when a field in a record with an originally failing explicit edit is changed. *Lemma* 1 gives a formulation on how to generate implicit edits.

Lemma 1 (Fellegi and Holt [1976]): If E^r are edits $\forall\ r \in S$, where S is any index set,

$$E^r : \bigcap_{j=1}^{n} \boldsymbol{A}_j^r = F, \quad \forall \quad r \in S.$$

Then, for each i $(1 \leq i \leq n)$, the expression

$$E^* : \bigcap_{j=1}^{n} \boldsymbol{A}_j^* = F \tag{2}$$

is an implied edit, where

$$\boldsymbol{A}_j^* = \bigcap_{r \in S} \boldsymbol{A}_j^r \neq \emptyset \quad j = 1, \cdots, i-1, i+1, \cdots, n$$

$$\boldsymbol{A}_i^* = \bigcup_{r \in S} \boldsymbol{A}_i^r \neq \emptyset.$$

If all the sets A_i^r are proper subsets of A_i, i.e., $A_i^r \neq A_i$ (field i is an entering field of edit E^r) $\forall\ r \in S$, but $A_i^* = A_i$, then the implied edit (2) is called an *essentially new edit.* Field i, which has n_i possible values, is referred to as the *generating field* of the implied edit. The edits $E^r\ \forall\ r \in S$ from which the new implied edit E^* is derived are called *contributing edits.*

Therefore, in order to generate an essentially new implicit edit, we must have the following three conditions:

1. $A_j^* \neq \emptyset, \forall\ j,\ 1 \leq j \leq n$;

2. $A_i^r \neq A_i, \forall\ r \in S$, where $A_i^r \neq \emptyset$;

3. $A_i^* = A_i$.

Conditions 2 and 3 indicate that the set $\{A_i^r \mid r \in S\}$ is a cover of A_i and are the foundations of the following set covering formulation in (3).

Let $\{E^r \mid r \in S\}$ be the set of the s edits with field i entering, then the set covering problem related to the generating field i is

$$\begin{array}{rl} \text{Minimize} & \sum_{r \in S} x_r \\ \\ \text{subject to} & \sum_{r \in S} g_{jr}^i x_r \geq 1, \qquad j = 1, 2, \cdots, n_i \\ \\ & x_r = \begin{cases} 1, & \text{if } E^r \text{ is in the cover;} \\ 0, & \text{otherwise,} \end{cases} \\ & \quad r \in S \end{array} \tag{3}$$

where

$$g_{jr}^i = \begin{cases} 1, & \text{if } E^r \text{ contains the } j\text{th element in field } i; \\ 0, & \text{otherwise,} \end{cases}$$

is the jth element in field i of edit E^r $(r \in S)$. If $\boldsymbol{x}$ is a prime cover solution to (3) and $K = \{r \mid x_r = 1\} \subset S$, then $\cup_{k \in K} A_i^k = A_i$. A prime cover solution is a nonredundant set of the edits whose ith components cover all possible values of the entering field, which is the generating field to yield an essentially new implicit edit. The dimension of the constraint matrix $\boldsymbol{G}= (g_{jr}^i)_{n_i \times s}$ is $n_i \times s$. The size of the preorder forest of the set covering algorithm described in Chen [1998] is $2^s - 1$.

3 Formulation of Age Comparisons

We will describe the age comparisons with the study given in Chen, Thibaudeau, and Winkler [2002], in which the 1999 ACS (American Community Surveys) data set and *the 1999 ACS Edit and Allocation Specifications* are used. Suppose we have a survey with 4 questions of sex, age, household relationship, and marital status in each household. We also assume that each household contains at most three persons, in which the first person is the householder, the second person is the spouse if there is one. For the conversions of households with more than three persons, see Chen, Winkler, and Hemmig [2000].

To identify the explicit edits of DISCRETE system for this paper, we define the first nine fields (or variables) that are sex, household relationship, and marital status for the three members in a household: `SEXU11` (meaning the first person's sex), `RELANU11`, `MARSTU11`, `SEXU22`, `RELANU22`, `MARSTU22`, `SEXU33`, `RELANU33`, and `MARSTU33`, Table 1 lists the variable names and their possible coded values. The other fields are for the age comparison condition variables.

In the age comparison, each time when a new age restriction appears in one of the if-then-else rules in *the 1999 ACS Edit and Allocation Specifications*, an age comparison condition variable is defined as described in Chen, Winkler, and Hemmig [2000]. An age comparison condition variable is an inequality of the form:

$$a_1 x_1 + a_2 x_2 + a_3 x_3 > b, \tag{4}$$

Table 1: All Possible Values for sex, hhr, and ms with DISCRETE.

sex	household relationship (hhr)	marital status (ms)
`SEXU11, SEXU22, SEXU33`	`RELANU11, RELANU22, RELANU33`	`MARSTU11, MARSTU22, MARSTU33`
1 = Male	1 = Householder	1 = Now married
2 = Female	2 = Husband/wife	2 = Widowed
3 = Unknown	3 = Son/daughter	3 = Divorced
	4 = Brother/sister	4 = Separated
	5 = Father/mother	5 = Never married
	6 = Grandchild	6 = Unknown
	7 = In-law	
	8 = Other relative	
	9 = Roomer/boarder	
	10 = Housemate/roommate	
	11 = Unmarried partner	
	12 = Foster child	
	13 = Other nonrelative	
	14 = Unknown	

where a_i $(i = 1, 2, 3)$ is one of the three values: -1, 0, and 1, and x_i is the ith person's age. There are two possible values for each of the age comparison condition variables: 1 if (4) is true; 2 if false. This type of variables is called *Boolean* age comparison condition variables. Table 2 lists the 41 Boolean age comparison condition variables of (4) described in Chen, Thibaudeau, Winkler [2002]. For example, one of the 41 Boolean age comparison condition variables is $x_1 - x_2 > -12$ (`VAR23`), where $a_1 = 1$, $a_2 = -1$, and $a_3 = 0$. If the first person's age is 35 and the second is 32, then the value of the variable of $x_1 - x_2 > -12$ is 1 because it is true that $35 - 32 > -12$. Another example is that the first person's age is less than or equal to 14: $x_1 \leq 14$, that is converted to the normalized form of $-x_1 > -15$ in (4) with $a_1 = -1$, $a_2 = a_3 = 0$, and $b = -15$ (`VAR10`).

Table 2: The 41 Boolean Age Comparison Condition Variables.

Field ID	a_1	a_2	a_3	b	Field ID	a_1	a_2	a_3	b	Field ID	a_1	a_2	a_3	b
`VAR10`	−1	0	0	−15	`VAR24`	1	0	−1	−12	`VAR38`	1	0	−1	−15
`VAR11`	0	−1	0	−15	`VAR25`	−1	1	0	−30	`VAR39`	0	−1	0	−30
`VAR12`	0	0	−1	−15	`VAR26`	−1	0	1	−30	`VAR40`	0	0	−1	−30
`VAR13`	1	0	0	115	`VAR27`	0	−1	0	−18	`VAR41`	0	1	0	59
`VAR14`	0	1	0	115	`VAR28`	0	0	−1	−18	`VAR42`	0	0	1	59
`VAR15`	−1	1	0	−12	`VAR29`	1	−1	0	0	`VAR43`	−1	0	1	−20
`VAR16`	1	−1	0	49	`VAR30`	0	1	−1	14	`VAR44`	0	−1	1	−20
`VAR17`	−1	0	1	−12	`VAR31`	1	0	−1	14	`VAR45`	0	−1	1	−12
`VAR18`	1	0	−1	49	`VAR32`	0	1	0	74	`VAR46`	0	1	0	89
`VAR19`	1	−1	0	34	`VAR33`	−1	1	0	−15	`VAR47`	0	0	1	89
`VAR20`	−1	1	0	34	`VAR34`	0	0	1	74	`VAR48`	−1	0	0	−30
`VAR21`	1	0	−1	34	`VAR35`	−1	0	1	−4	`VAR49`	−1	0	1	−25
`VAR22`	−1	0	1	34	`VAR36`	0	−1	1	−4	`VAR50`	0	−1	1	−25
`VAR23`	1	−1	0	−12	`VAR37`	0	1	−1	−15					

With the 41 Boolean age comparison condition variables combined with the first nine variables, we identified 83 explicit edits from *the 1999 ACS Edit and Allocation Sepcifications.*

The following example illustrates how the Boolean age comparison variables are used to identify the edit rule of a householder's age being less than 15: $A_2^o = \{1\}$ (RELANU11) and $A_{10}^o = \{1\}$ (VAR10). The normal form of the edit is

$$A_1 \times \{1\} \times A_3 \times \cdots \times A_9 \times \{1\} \times A_{11} \times \cdots \times A_{50} = F \tag{5}$$

Another example is $A_5^o = \{3\}$ (RELANU22), $A_{32}^o = \{1\}$ (VAR32), and $A_{33}^o = \{1\}$ (VAR33), in which the second person's household relationship (RELANU22) is child, the age (VAR32) is greater than 74, and the first person is less than 15 years older than the second person. In this example, the if-then-else edit rules in *the 1999 ACS Edit and Allocation Specifications* are given in Table 3. The normal form of the edit is

Table 3: Example of If-Then-Else Rules.

Universe	*Child with Age* is greater than or equal to 75;
If. . .	*Age of Reference person* − *Age* is less than 15 and *Marital status* = Never married or SAS missing;
Then. . .	Blank *Age*; tally Z(12); set allocation flag.
Universe	*Child with Age* is greater than or equal to 75;
If. . .	*Age of Reference person* − *Age* is less than 15 and *Marital status* = Ever married;
Then. . .	Blank *Relationship*; tally Z(13); set allocation flag.

$$A_1 \times \cdots \times A_4 \times \{3\} \times A_6 \times \cdots \times A_{31} \times \{1\} \times \{1\} \times A_{34} \times \cdots \times A_{50} = F \tag{6}$$

The third example is is $A_5^o = \{3\}$ (RELANU22), $A_{15}^o = \{1\}$ (VAR15), and $A_{46}^o = \{1\}$ (VAR46), in which the second person's household relationship (RELANU22) is child, the age (VAR46) is greater than 89, and the first person is less than 12 years older than the second person. The normal form of the edit is

$$\begin{aligned} A_1 \times \cdots \times A_4 \times \{3\} \times A_6 \times \cdots \times A_{14} \times \{1\} \times A_{16} & \\ \times \cdots \times A_{45} \times \{1\} \times A_{47} \times \cdots \times A_{50} = F & \end{aligned} \tag{7}$$

This example is a redundant edit because if a record failing (7) must fail (6). However, edits (6) and (7) are two separate edits and are not dominated by each other with the 41 Boolean age comparison condition variables because one of them is not a subset of the other.

The age comparison also identifies 695 explicit edits, each of which is a contradiction condition within a subset of the 41 Boolean age comparison condition variables. For example, the normal form of the explicit edit

$$\begin{aligned} A_1 \times \cdots \times A_9 \times \{2\} \times \{1\} \times A_{12} \times \cdots \times A_{19} \times \{1\} & \\ \times A_{21} \times \cdots \times A_{50} = F & \end{aligned} \tag{8}$$

with $A_{10}^o = \{2\}$ (VAR10), $A_{11}^o = \{1\}$ (VAR11), and $A_{20}^o = \{1\}$ (VAR20) defines a contradiction situation among the variables VAR10, VAR11, and VAR20. If this edit is rewritten as the following inequalities:

$$\begin{array}{lrrrrrrl} \texttt{VAR10:} & - & x_1 & & & \leq & - & 15 \\ \texttt{VAR11:} & & & - & x_2 & > & - & 15 \\ \texttt{VAR20:} & - & x_1 & + & x_2 & > & & 34 \end{array} \tag{9}$$

it is clear that there are no values for x_1 (the first person's age) and x_2 (the second person's age) to satisfy the above three inequalities.

The implementation of the Boolean age comparison condition variables is very inefficient because the values of m, n in (1) and n_i, s in (3) are much larger than necessary. These large values provide a major contribution to the inefficient operations of the set covering algorithm to solve (1) and (3). The alternative approach we are going to describe below will reduce significantly the sizes of the constraint matrices in (1) and (3) and therefore reduce a great amount of computation effort to solve (1) and (3).

Instead of using the Boolean age comparison condition variables, we will use the categorical age comparison variables. For three-person households, there are at most 6 age comparisons: 3 *within person age comparisons* and 3 *between persons age comparisons*. They are `AGEU10`, `AGEU11`, `AGEU12`, `AGEU13`, `AGEU14`, and `AGEU15`. If x_i is the ith person's age, then the 6 variables have the form:

$$a_1x_1 + a_2x_2 + a_3x_3, \tag{10}$$

where (a_1, a_2, a_3) is one of the following triples: (0, 0, 1), (0, 1, 0), (0, 1, -1), (1, 0, 0), (1, 0, -1), and (1, -1, 0). The six variables are then fit to the Fellgi-Holt model described in Section 2. For example, the eight Boolean variables, `VAR15`, `VAR16`, `VAR19`, `VAR20`, `VAR23`, `VAR25`, `VAR29`, and `VAR33` in Table 2 can be combined into the form of (10) with (a_1, a_2, a_3) = (1, -1, 0). The new created categorical variable $x_1 - x_2$ (`AGEU15`) will take nine coded values as its valid values. Tables 4 and 5 illustrate how the 9 coded values are obtained for the categorical variable $x_1 - x_2$. First, the 8 inequalities are revised such that the leading coefficient of the left hand side, a_1, is equal to 1 as given in Table 4. Then, the 8 revised inequalities are sorted in increasing order according to the values of the right hand side, b, as given in Table 5. Each of the 8 Boolean variables has two Boolean values, `true` or `false`. The Boolean values are represented by integer intervals. For example, in the first Boolean variable `VAR20` in Table 5, the integer values of $x_1 - x_2$ in $L_{20} = [-999, -35]$ make

Table 4: The Categorical Variable $x_1 - x_2$ (A).

Variable	Original Inequality	Revised Inequality
`VAR15`	$-x_1 + x_2 > -12$	$x_1 - x_2 < 12$
`VAR16`	$x_1 - x_2 > 49$	$x_1 - x_2 > 49$
`VAR19`	$x_1 - x_2 > 34$	$x_1 - x_2 > 34$
`VAR20`	$-x_1 + x_2 > 34$	$x_1 - x_2 < -34$
`VAR23`	$x_1 - x_2 > -12$	$x_1 - x_2 > -12$
`VAR25`	$-x_1 + x_2 > -30$	$x_1 - x_2 < 30$
`VAR29`	$x_1 - x_2 > 0$	$x_1 - x_2 > 0$
`VAR33`	$-x_1 + x_2 > -15$	$x_1 - x_2 < 15$

Table 5: The Categorical Variable $x_1 - x_2$ (B).

Boolean Variable	Revised Inequality	Boolean Values		Closed Interval	Coded Value
		True	False		
VAR20	$x_1 - x_2 < -34$	$L_{20} = [-999, -35]$	$U_{20} = [-34, 999]$	$L_{20} = [-999, -35]$	1
VAR23	$x_1 - x_2 > -12$	$U_{23} = [-11, 999]$	$L_{23} = [-999, -12]$	$U_{20} \cap L_{23} = [-34, -12]$	2
VAR29	$x_1 - x_2 > 0$	$U_{29} = [1, 999]$	$L_{29} = [-999, 0]$	$U_{23} \cap L_{29} = [-11, 0]$	3
VAR15	$x_1 - x_2 < 12$	$L_{15} = [-999, 11]$	$U_{15} = [12, 999]$	$U_{29} \cap L_{15} = [1, 11]$	4
VAR33	$x_1 - x_2 < 15$	$L_{33} = [-999, 14]$	$U_{33} = [15, 999]$	$U_{15} \cap L_{33} = [12, 14]$	5
VAR25	$x_1 - x_2 < 30$	$L_{25} = [-999, 29]$	$U_{25} = [30, 999]$	$U_{33} \cap L_{25} = [15, 29]$	6
VAR19	$x_1 - x_2 > 34$	$U_{19} = [35, 999]$	$L_{19} = [-999, 34]$	$U_{25} \cap L_{19} = [30, 34]$	7
VAR16	$x_1 - x_2 > 49$	$U_{16} = [50, 999]$	$L_{16} = [-999, 49]$	$U_{19} \cap L_{16} = [35, 49]$	8
				$U_{16} = [50, 999]$	9

the inequality $x_1 - x_2 < -34$ true and the values in $U_{20} = [-34, 999]$ make it false. Note that $L_{20} \cap U_{20} = \emptyset$. The numbers -999 and 999 represent an open end of intervals. The coded value 1 represents the interval L_{20} because every integer in L_{20} is less than that in U_{20}, L_{20} is called the lower interval and U_{20} the upper interval. Coded value 2 represents $U_{20} \cap L_{23} = [-34, -12]$, which is the intersection of VAR20's upper interval and VAR23's lower interval. Similarly, coded values 3 to 8 represent their respective intersections. Coded value 9 represents the last Boolean variable VAR16's upper interval. In other words, the number of coded values is one more than the number of Boolean variables with the same form of $a_1x_1 + a_2x_2 + a_3x_3$.

Table 6 lists the six categorical valiables and their possible coded values converted from the 41 Boolean age comparison condition variables of (4) listed in Table 2. Each coded value represent a closed interval with integers. This formulation significantly reduced the size of the set covering problem of the edit generation and the error localization.

If the six categorical age comparison variables are used, the normal form of the edit (5) becomes

$$A_1 \times \{1\} \times A_3 \times \cdots \times A_{12} \times \{1\} \times A_{14} \times A_{15} = F \tag{11}$$

with $A_2^o = \{1\}$ (RELANU11, the first person is the householder) and $A_{13}^o = \{1\}$ (VAR10, the first person's age is less than 15), in which VAR10 becomes part of AGEU13. The normal form of edit (6) becomes

$$\begin{aligned} A_1 \times \cdots \times A_4 \times \{3\} \times A_6 \times \cdots \times A_{10} \times \{6,7,8\} \times A_{12} & \\ \times A_{13} \times A_{14} \times \{1,2,3,4,5\} = F & \qquad (12) \end{aligned}$$

with $A_5^o = \{3\}$ (RELANU22, the second person is a child), $A_{11}^o = \{6,7,8\}$ (AGEU11, the second person's age is greater than 74), and $A_{15}^o = \{1,2,3,4,5\}$ (AGEU15, the first person is at most

Table 6: The Six Categorical Variables Defined for Age Comparisons.

Variable Name	Form (10) (a_1, a_2, a_3)	Coded Values	Variable Name	Form (10) (a_1, a_2, a_3)	Coded Values
AGEU10	x_3 (0,0,1)	1 = [0, 14] 2 = [15, 17] 3 = [18, 29] 4 = [30, 59] 5 = [60, 74] 6 = [75, 89] 7 = [90, 999] 8 = unknown*	AGEU13	x_1 (1,0,0)	1 = [0, 14] 2 = [15, 29] 3 = [30, 115] 4 = [116, 999] 5 = unknown*
AGEU11	x_2 (0,1,0)	1 = [0, 14] 2 = [15, 17] 3 = [18, 29] 4 = [30, 59] 5 = [60, 74] 6 = [75, 89] 7 = [90, 115] 8 = [116, 999] 9 = unknown*	AGEU14	$x_1 - x_3$ (1,0,−1)	1 = [−999, −35] 2 = [−34, −15] 3 = [−14, −12] 4 = [−11, 3] 5 = [4, 11] 6 = [12, 14] 7 = [15, 19] 8 = [20, 24] 9 = [25, 29] 10 = [30, 34] 11 = [35, 49] 12 = [50, 999] 13 = unknown*
AGEU12	$x_2 - x_3$ (0,1,−1)	1 = [−999, −15] 2 = [−14, 3] 3 = [4, 11] 4 = [12, 14] 5 = [15, 19] 6 = [20, 24] 7 = [25, 999] 8 = unknown*	AGEU15	$x_1 - x_2$ (1,−1,0)	1 = [−999, −35] 2 = [−34, −12] 3 = [−11, 0] 4 = [1, 11] 5 = [12, 14] 6 = [15, 29] 7 = [30, 34] 8 = [35, 49] 9 = [50, 999] 10 = unknown*
* *if the age of at least one of the involved person(s) is unknown or invalid*					

14 years older than the second person). The normal form of edit (7) becomes

$$\begin{aligned} A_1 \times \cdots \times A_4 \times \{3\} \times A_6 \times \cdots \times A_{10} \times \{7, 8\} \times A_{12} & \\ \times A_{13} \times A_{14} \times \{1, 2, 3, 4\} = F & \end{aligned} \tag{13}$$

with $A_5^o = \{3\}$ (RELANU22, the second person is a child), $A_{11}^o = \{7, 8\}$ (AGEU11, the second person's age is greater than 89), and $A_{15}^o = \{1, 2, 3, 4\}$ (AGEU15, the first person is at most 11 years older than the second person). It is very clear that (13) is a subset of (12) and therefore is a redundant edit. It will not be included in the input edit table. The normal form of edit (8) becomes

$$A_1 \times \cdots \times A_{10} \times \{1\} \times A_{12} \times \{2, 3, 4\} \times A_{14} \times \{1\} = F \tag{14}$$

which is one (or subset of one) of the 67 identified explicit edits, each of which is a contradiction condition within a subset of the six categorical age comparison variables.

4 The Performance Measurement

The performance of the two formulations of age comparisons: Boolean variables and categorical variables are illustrated with the 1999 ACS data set and *the 1999 ACS Edit and Allocation Specifications.* We ran both formulations to solve the integer programming of (3) using the set covering algorithm of Chen [1998] to generate all the implicit edits. The performance is given in Table 7. The processing time (measured on a Sun Ultra machine) of the generation of the implicit edits is significantly reduced with the categorical age comparison variables.

Table 7: The Perfromance of Boolean and Categorical Variables in Edit Generation.

	Boolean Variables	Categorical Variables
number of age comparison variables	41	6
number of other variables	9	9
number of identified explicit edits	83	74
number of age only explicit edits	695	67
total number of explicit edits	778	141
total number of explicit and implicit edits	9948	437
processing time (elapsed time, hr:mn:sc)	26:49:45	00:01:18

Another measurement of the performance is the size of the contraint matrix given in (1), which is to solve the error localization. The comparison of the sizes of the contraint matrices for 22 households of the 1999 data set is given in Table 8. The actual sizes are larger than those given in Table 8, which are after the reduction of the matrices and before the construction of the preorder forest for the set covering algorithm given in Chen [1998]. Also included in Table 8 are the sizes, $2^n - 1$, of the preorder forests to repeatedly solve the integer programming problem of (1).

5 Discussion and Summary

The formulation with Boolean variables is simple and all the Boolean age comparison condition variables have only two values (the value of "unknown" is excluded). However, we will pay a huge price of running the edit generation of (3) and the error localization of (1). For example, it will take forever for the set covering algorithm to complete a preorder forest of size $2^{28} - 1$ as illustrated in Table 8. The formulation with categorical variables simplifies the number of fields (variables) needed and the number of explicit and implicit edits required to solve the error localization. Although the categorical variables have more value states, the number of value states has nothing to do with the performance of running the edit generation

Table 8: The Sizes of the Constraint Matrices of (1).

household	Boolean Variables			Categorical Variables		
record #	nrows (m)	ncols (n)	2^n-1	nrows (m)	ncols (n)	2^n-1
1	98	28	$2^{28}-1$	6	7	2^7-1
2	104	29	$2^{29}-1$	6	7	2^7-1
3	110	30	$2^{30}-1$	6	7	2^7-1
4	85	28	$2^{28}-1$	6	7	2^7-1
5	85	28	$2^{28}-1$	6	7	2^7-1
6	66	27	$2^{27}-1$	6	7	2^7-1
7	47	24	$2^{24}-1$	6	7	2^7-1
8	82	28	$2^{28}-1$	6	6	2^6-1
9	85	28	$2^{28}-1$	6	7	2^7-1
10	85	28	$2^{28}-1$	6	7	2^7-1
11	85	28	$2^{28}-1$	6	7	2^7-1
12	20	13	$2^{13}-1$	9	8	2^8-1
13	44	28	$2^{28}-1$	10	9	2^9-1
14	93	28	$2^{28}-1$	8	8	2^8-1
15	59	26	$2^{26}-1$	6	6	2^6-1
16	65	27	$2^{27}-1$	6	6	2^6-1
17	98	28	$2^{28}-1$	6	7	2^7-1
18	36	24	$2^{24}-1$	6	6	2^6-1
19	36	24	$2^{24}-1$	6	6	2^6-1
20	36	24	$2^{24}-1$	6	6	2^6-1
21	52	24	$2^{24}-1$	6	7	2^7-1
22	52	24	$2^{24}-1$	6	7	2^7-1

and error localization because the implementaion used in the set covering algorithm is the bitwise operations on a 32-bit word.

In the age comparisons between members of a household, the number of Boolean variables, b, may grow as large as the number of age restrictions in the edit specifications. When the number of age restrictions becomes very large, it is impractical to use the Boolean age comparison condition variables. In contrast, the number of categorical age comparison variables has an upper bound. For k-person households, the upper bound, c, is

$$c = \binom{k}{1} + \binom{k}{2} = \binom{k+1}{2} = \frac{k(k+1)}{2}. \tag{15}$$

Therefore, the performance improvement of using the categorical age comparison variables becomes enormous when b becomes very large. The new formulation of age comparisons makes the DISCRETE edit system applicable to a wider range of surveys that require age comparisons for statistical editing.

References

[1] B. Chen. Set covering algorithms in edit generation. In *Proceedings of the Statistical Computing Section*, pages 91–96. American Statistical Association, 1998.

[2] B. Chen, Y. Thibaudeau, and W. E. Winkler. A Comparison Study of ACS If-Then-Else, NIM, and DISCRETE Edit and Imputation Systems Using ACS Data. Research Report *forthcoming*, Statistical Research Division, Bureau of the Census, Washington, D.C., 2002.

[3] B. Chen, W. E. Winkler, and R. J. Hemmig. Using the DISCRETE Edit System for ACS Surveys. Research Report RR2000/03, Statistical Research Division, Bureau of the Census, Washington, D.C., 2000.

[4] I. P. Fellegi and D. Holt. A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association*, 71:17–35, 1976.

[5] R. S. Garfinkel, A. S. Kunnathur, and G. E. Liepins. Optimal imputation of erroneous data: Categorical data, general edits. *Operations Research*, 34:744–751, 1986.

[6] W. E. Winkler. Set-covering and editing discrete data. Technical report, Bureau of the Census, 1997.

[7] W. E. Winkler and T. F. Petkunas. The DISCRETE Edit System. Statistical Research Division Research Report, Bureau of the Census, 1996.