STATE OF STATISTICAL DATA EDITING AND CURRENT RESEARCH PROBLEMS

William E. Winkler, bwinkler@census.gov

1. INTRODUCTION

This paper is my description of the state of statistical data editing and current research problems. It is not intended to be a complete description of all areas. Rather, it represents sub-areas of statistical data editing that I will describe in sufficient detail so that the discussion of a few research problems is more easily understood.

I define statistical data editing (SDE) as those methods that are used to edit (i.e., clean-up) and impute (fill-in) missing or contradictory data. The end result of SDE is data that can be used for intended analytic purposes. These include primary purposes such as estimation of totals and subtotals for publications that are free of self-contradictory information. The published totals do not contradict published totals in other sources. Self-contradictory information might include groups of items that do not add to desired subtotals or totals for subgroups that exceed a known proportion of the total for the entire group. The uses of the data after SDE might be preparation of variances of estimates for a number of sub-domains and micro-data analyses. If only a few published totals need to be accurate, then an efficient use of resources may be to perform detailed edits on only a few records that effect the estimated totals. If many analyses need to be performed on a large number of sub-domains or if the full set of accurate micro-data are needed, then a very large number of edits, follow-up, and corrections may be needed.

SDE can be used in all phases of survey processing. These phases include frame development, form design, proposed analytic purposes for which the data are collected, and quality assurance. This paper focuses primarily on SDE as it applies to analytic purposes, and places most emphasis on those procedures typically applied after the initial receipt of survey or other data. The main goal of SDE might be improved procedures and greater automation to enhance the ability of survey managers and analysts to provide accurate published estimates and micro-data.

I broadly subdivide statistical data editing into two subcategories: (1) Fellegi-Holt (FH) methods and systems and (2) General methods and systems. FH systems are based on the Fellegi-Holt model of editing and typically add various options for imputation. General methods are all other methods. Whereas the paper by Fellegi and Holt (1976) appeared quite awhile ago, few systems have been implemented because of the difficulty in developing expertise in the operations research (OR) techniques needed under the model. Many statistical agencies have chosen to concentrate on traditional methods that are a large sub-portion of general methods. These traditional methods include if-then-else rules for detecting contradictory information and various ways of imputing values of variables to replace the contradictory values. The dilemma with if-then-else rules is that they may not be straightforward to develop and may be difficult to write into computer code. If there are slight changes in the survey form and edit rules, then subsets of thousands of lines of code may need to be rewritten and debugged. The reason that FH methods are so appealing is that most of the if-then-else types of edits can be put in tables that are straightforward to modify and update. Because the source code does not need any updating, it is possible to create a FH system for editing that can be developed and maintained for different surveys by non-programmers such as subject matter specialists, statisticians, and economists.

General methods include selective (or macro) editing. In some situations, the records associated with the largest companies or firms can be delineated for follow-up and review. Some of these methods use measures such as the Hidiroglu-Berthelot (HB) statistic or graphical displays. HB methods involve measures that determine which records cause the largest deviations of key totals in the survey population.

Follow-up is more efficient because the most important records are reviewed first. DesJardins (1997, 1998) has recently developed courses for using point-and-click graphical methods (via SAS Insight or JMP) as exploratory data analysis (EDA) tools for finding erroneous data. The significant advantage of EDA methods is that they allow non-programmers to delineate and review data in ways that are different from the fixed ways available in a specific edit system for an individual survey. For more information about general editing methods, see Pritzker et al. (1965).

The outline of this paper is as follows. Section II provides background Fellegi-Holt systems and methods as they are currently implemented in software systems at statistical agencies. In Section III, I describe a few subtopics of the FH methods more fully and cover selected research problems. The final section gives concluding remarks.

2. FELLEGI-HOLT METHODS AND SYSTEMS

In many situations, data files are edited using custom software that incorporates if-then-else rules developed by subject-matter specialists. If the specialists are unable to develop the full logic needed for the edit rules, then the subsequent edit software can be in error. If programmers do not properly code the rules, then the software would be in error. Developing software from scratch each time a survey form and database is redesigned is time-consuming and error-prone. It is better to have a system that can describe edit rules in tables that are read and utilized by reusable software modules.

Fellegi and Holt (1976), hereafter referred to as FH, reasoned that it is better to create a system that has edit rules in tables that are read and utilized by reusable software modules. The tables could be more easily updated and maintained than complex if-then-else rules in computer code. FH gave methods showing how easily general if-then-else edit rules could be translated into a form used in the tables. The software would automatically check the logical validity of the entire system prior to the receipt of data during production processing. Checking the logical validity is often referred to as determining the *consistency* or *logical consistency* of a set of edits. If a set of edits is inconsistent, then there exist no records that can satisfy the set of edits. FH provided the theoretical basis of such a system that had three goals that (paraphrased) are:

1. The data in each record should be made to satisfy all edits by changing the fewest possible variables (fields).
2. Imputation rules should derive automatically from edit rules.
3. When imputation is necessary, it should maintain the joint distribution of variables.

The key to the FH approach is to understand the underpinnings of goal 1. Goal 1 is referred as the *error localization* problem. To solve the error-localization problem, FH showed that both explicit and implicit edits are needed. *Implicit* edits are those that can be derived (or generated) from a set of explicit edits. If the implicit edit fails, then necessarily at least one of the explicit edits used in generating the implicit edit fail. In work prior to the FH paper, many authors introduced edit methods for identifying fields (variables) to change that could not assure that the changed record satisfied all edits. As proved by FH, implicit edits can provide information about explicit edits that do not fail with the original un-edited record but might fail if information in the implicit edit is not properly used. The main theorem in FH is a landmark result because it demonstrated that it is always possible to find a set of fields to change in a record that yield a changed record that satisfies all edits. FH's proof is by an inductive, existence-type method that does not give insight into how to deal with practical computational aspects of generating implicit edits. A well-implemented FH system can yield substantial efficiencies because it dramatically reduces the need for highly skilled programmers and subject-matter specialists to develop a set of logically consistent edits. The FH ideas give formal ways of development that greatly facilitate creating sets of edits. The key features of a Fellegi-Holt system are:

1. Edit restraints reside in easily modified tables.

2. The logical consistency of the entire edit system can be checked prior to the receipt of data.
3. The main logic resides in reusable mathematical routines.
4. In one pass through the system, records satisfy edits.

Implementations of FH systems have typically either been for discrete data (e.g., categorical) to which arbitrary edits are applied or for continuous data to which ratio or linear inequality edits are applied. To understand why discrete and continuous data implementations are so different, I provide elementary examples. A survey form for a demographic survey might contain the following information. If variables take a finite number of values (typically a small number), then the FH system is called discrete. In the following, it is likely that the marital status of the six-year-old son is likely to be in error. Imputing a new value (or value-state) to marital status of 'Single' and the resultant imputed record would satisfy an edit that states that a record is in error if an individual is less than 16 years of age and is married. If-then-else types of edits for discrete data are easily converted to a form E^1 ={age < 16, Marital_Status = Married} which effectively states that an error condition occurs if a record has a sub-component that takes a value in edit set E^1.

Example-Discrete

Age	*Race*	*Sex*	*Marital Status*	*Relation-ship*
35	1	M	Married	Head
6	1	M	Married	Son

Impute Marital_Status Married - > Single

Whereas it is possible for an individual who is less than 16 to be married, in the overwhelming majority of situations, a record that satisfied E^1 would truly be in error. Application of this edit is necessarily a compromise. In a few situations, changing the marital status of a person who has age less than 16 to unmarried might induce an error. In most situations, the change would be a correction. With larger survey forms, there may be very complicated relationships or skip patterns that can make the application more complicated. As individuals gain experience with the most elementary forms of the edits, then they are also able to develop edits for most of the most complicated relationships. Within the framework of a given general edit system, it is not possible (or at least easily possible) to replace all specific edits in a non-general system for an individual survey. A goal might be to replace 90-95% of all the predecessor system with the same set of edits in the form that the new generalized system accepts. In some situations, the 5-10% of the remaining predecessor-system edits may not be needed in the application of the general system. In other situations, it may be possible to adjoin a few of the 5-10% in a module to the general system.

To make ideas more precise for determining the need for implicit edits, it is possible to define an edit as a set of points. For instance, edit E^1 = {married & age < 16}. If $r \in E^1$, then r fails edit E^1. Consider the following edits:

E^1 = {age < 16, married},
E^2 = {not married, spouse}, and
E^3 = {age < 16, spouse}.

Edits E^1 and E^2 imply E^3 (i.e., E^1 & E^2 => E^3). If edit E^3 fails, then necessarily either E^1 fails or E^2 fails. If edits E^1 and E^2 are explicitly defined by survey specialists, then edit E^3 is an *implicit edit* generated by E^1 and E^2. If a record fails edit E^1 and does not fail edit E^2, then edit E^3 (which must also fail) gives information that allows determination of value-states of fields so that the resultant changed record satisfies all edits. For instance, if record r fails edit E^1 and only the explicit edit E^1 is used, then a solution

for the field to change would be marital_status. If marital_status is changed from 'married' to 'not married', then record r would fail edit E^2. In other words, the explicit edit E^3 gives information about non-failing explicit edits that is needed so that the changed record r satisfies all edits. FH provided a method for generating all implicit edits. They gave a proof that their method generated all the required implicit edits needed for error localization. The FH generation method involved several components. First, a field is chosen on which to generate (the *generation field*) and a set of edits is chosen (the *generating set*). The field in the candidate implicit edit corresponding to the generating field consists of the union of all value states for the field that are taken from the generating edits. The remaining fields in the candidate implicit edit are obtained by taking the intersection of all remaining fields where the intersection of over the value-states in the generating set of edits being generated on. If the union in the generating field is not all value-states, then the candidate implicit edit is dropped for further consideration. If the intersection of any of the fields in the candidate implicit edit is null, then the implicit edits is also dropped. For the above example, edit E^3 is generated on the field marital_status. Edit E^3 places no restriction on the value-states of the marital_status field.

With continuous data the situation is similar. If two edits, take the form: E^1: $x_1 - x_2 < b$ and E^2: $x_2 + x_2 < c$, then edit E^3: $a_1 x_1 + a_3 x_3 < d$ can be generated from E^1 and E^2 by eliminating field x_2. More generally, it is possible to define edits by functions F_1 and F_2 on variables x_1, x_2, and x_3 where E^1: $F_1 < b$ places restrictions on x_1 and x_2 and E^1: $F_2 < c$ places restrictions on x_2 and x_3. If the functions $F_1$and F_2 can be solved to eliminate x_2, then a new function F_3 on x_1 and x_3 is produced (generated). The function F_3 is an implicit edit that places no restrictions on x_2. Ratio edits of the form $L_{ij} < V_i / V_j < U_{ij}$ are a special case of linear-inequality edits where U_{ij} is an upper bound and L_{ij} is a lower bound on variables V_i and V_j. The first inequality $L_{ij} < V_i / V_j$ is equivalent to the linear-inequality edit E^4: $L_{ij} V_j < V_i$.

In practice, general algorithms for determining all implicit edits have only been developed for sets of linear inequality edits and for sets of ratio edits. Generating implicit edits for ratio edits is particularly easy. If there are n fields involved in ratio edits, then there can exist at most n(n-1)/2 ratio edits (explicit and implicit). With linear inequality edits, the number of implicit edits can increase dramatically. For instance, with 30 explicit edits and 10 fields, it might be possible to generate 400 implicit edits. Although the generation algorithm for linear inequality edits is straightforward to program, the algorithm is not particularly efficient. The generation can take 24 hours on a moderately fast computer with a moderate number of linear inequality edits.

Gordon Sande (1979) introduced a clever method to determine the set of solutions to the error-localization problem without generating all implicit linear inequality edits. Sande's ideas were particularly useful because the computers of twenty years ago were less than $1/200^{th}$ as fast as they are now. Sande used Chernikova's algorithm (1964, 1965) to generate the vertexes of a bounded region in R^n that was determined by the explicit linear inequality edits. The maximum and minimum solutions were known to occur on the vertexes. Because the basic Chernikova algorithm was too slow, Sande used a cardinality-restrained vertex generation algorithm due to D. S. Rubin (1975). Later, theoretical and heuristic work (Schopiu-Kratina and Kovar 1989, Filion and Schopiu-Kratina 1993) generally gave slight speed improvements. One heuristic modification to the basic algorithms gave a 60-fold speed improvement that is incorporated in the current version of Statistics Canada's Generalized Edit and Imputation System (GEIS). Statistics Canada's version of the Chernikova algorithm is also incorporated in the AGGIES system (Todaro 1998, 1999) which is written in SAS. Statistics Netherlands has independently developed similar variants of Chernikova's algorithm that are in CherryPi (DeWaal 1996, 1997) and are being incorporated in Blaise (Pergamentsev 1998, DeWaal 1998).

FH methods show their power with small demographic surveys. If a small survey has reasonably well-documented edits (or at least well understood), then it is possible for a non-programmer to create the tables of edits and effectively create a production edit system in less than one day. With larger, much more complicated demographic surveys, some conversions are often needed to put data in a form that would allow it to be edited by a FH system. Sometimes, the data and edits may need to be partitioned into subsets that are run separately during implicit edit generation and together during production editing. This can assure that the slow, edit-generation, part of the FH system can be run in a reasonable amount of

time. At other times, moderately sophisticated data conversions may be needed. For a test system being evaluated for the U.S. Decennial Census, extra variables are created that allow easier running of the main FH system. To perform the edit E = {person1=house_holder, person2=child, age_child ≥ age_householder + 15}, a new yes/no variable is created by software that represents the comparison of the two ages given above. In other words, two age variables and all the appropriate combinations of ages associated with the edit are replaced by the yes/no variable. The new variables complicate use of the FH system but are very convenient for tracking all the different combinations of ages of householders and children that would be in error. Implicit edits are generated for all explicit edits (including the newly induced edits described above) and error localization is based on all implicit edits and all variables. Although the main edit program automatically determines the minimum number of fields to impute (i.e., error localization), the minimum is in terms of the original variables and the induced variables. A final software algorithm makes a conversion to the original set of variables that may not be minimal in the sense of FH theory but is guaranteed to yield a solution.

3. SELECTED RESEARCH PROBLEMS

The research problems fall into two categories: (1) algorithms that improve the speed of the software and (2) adjunct software that provides analyses and outputs needed by other parts of the edit system.

3.1 **Speed Improvements**

The speed improvements are needed for two basic facets of FH systems: (1) implicit-edit generation via set covering algorithms and (2) error-localization in the main edit program. If implicit-edits are generated prior to editing, then the amount of computation needed for error localization can be significantly reduced. The reduction is so significant that the speed of the main edit program is no longer an issue. If implicit edits are not generated prior to editing, then the edit program will need to generate additional information that may be thought of as associated with the precise set of implicit edits that fail for each given record.

For generating implicit edits for discrete editing, the current most general algorithms are likely to be in the SCIA system (Barcaroli et al. 1997). These algorithms have limitations because they appear to need as much as 24 hours to generate implicit edits when 250 or more explicit edits are used. With a large survey form or with complicated edit situations, as many as 750 explicit edits may be needed. Because the amount of computation needed for generation grows at a very high exponential rate in the number of edits, it is unlikely that current algorithms can generate the full set of implicit edits with as many as 300 explicit edits. More specifically, the amount of computation for 250 explicit edits is of the order exp(exp(250)) and the amount of computation for 300 is exp(exp(300)). If all the implicit edits can not be generated for a given set of explicit edits, then one practical approach is to divide the set of explicit edits into subsets that are sufficiently small so that edit-generation can be accomplished. The disadvantage of this approach is that moderately sophisticated ways of dividing the original full set of explicit edits into the subsets of explicit edits may be needed. This places additional burden on the users of the system in terms of extra methods and programming for tracking the subsets. If subsets are used to generate implicit edits, then not all implicit edits can be obtained. It will not be possible to find error-localization solutions for some edit-failing records. Statistics Canada (with CANEDIT), the statistical office in Spain (with DIA), and ISTAT (with SCIA) have all had to partition the set of explicit edits for some large surveys.

If implicit edits are generated, then the main edit programs such as SCIA (Barcaroli et al. 1997) and DISCRETE (Petkunas and Winkler 1997, Winkler 1997) are easily fast enough for error localization. Because of this, one direction of research is to develop new set covering algorithms for implicit-edit generation that are orders of magnitude faster than currently existing algorithms. Winkler (1998) introduced new algorithms that can be more than 100 times as fast as previously existing algorithms in a limited number of situations. The algorithms drastically reduce overall computation by tracking specific computational paths during the first part of the computation when the first set of implicit edits are

computed from explicit edits only. The algorithm makes assumptions that what is learned during the first stages can be used in generating edits at all subsequent stages. For several situations in which skip patterns were not present on the survey form, this edit-generation algorithm appeared to work well and generate all edits. For situations with complicated skip patterns such as Italian Labour Force data, the algorithm failed to generate all implicit edits (possibly as many as 10%). The reason for the edit-generation failure was that skip patterns necessitate tracking the details of computation over multiple levels of the overall computation. The much slower, brute-force algorithms of SCIA (and of earlier DIA and CANEDIT) do generate edits correctly. They do not attempt to eliminate most computational paths through sophisticated algorithms that track many detailed sub-portions of the overall computation.

To improve edit generation further, Chen (1998) introduced more sophisticated algorithms than those used by Winkler. Chen obtains drastic speed improvements (a factor of 100) because he only generates prime covers instead of all covers as in Winkler (1998). To perform such sophisticated computation, he introduced new metrics for ordering the edits and eliminating most computation paths. In the algorithm of Winkler, most edits that were generated at the first stage of the computation were not maximal implicitly new edits (in the terminology of FH and Garfinkel, Kunnathur, and Liepins 1986). The new maximal implicit edits are known as the only ones needed for error localization. In the Chen algorithm, most of the edits generated at the first stage are maximal implicit edits. Chen is investigating a complete set of algorithms to generate the entire set of implicit edits. He has introduced additional new metrics that reduce the amount of computation associated with comparing a newly generated implicit edit with the currently existing set. This improves upon Winkler (1998) who used simple sequential comparison methods. So far, however, expected speed improvements in the overall algorithms have not been achieved. Further investigation is being performed to determine which sub-components of the overall computation are taking the most amount of time. Chen has developed additional new ways of looking at the sub-components of the computation. Whereas the original Winkler code is in FORTRAN, the Chen code is in C++. The fundamental research problem is: Can the basic edit-generation algorithms of Barcaroli (1997) in SCIA or of Winkler (1998) or Chen (1998) for versions of DISCRETE be speeded up sufficiently to generate all or nearly all implicit edits for surveys ranging in size from moderate to large. Are completely new methods of edit-generation needed? Can methods of error-localization in the main edit program be developed without all implicit edits being known? Currently, the main edit systems for linear inequality edits are GEIS (Kovar et al. 1991) and AGGIES (Todaro 1998, 1999). SPEER (Draper and Winkler 1997) uses similar ideas for a more limited set of linear inequality edits. These systems are characterized by the fact that they generate additional information needed for error localization without knowing all implicit edits.

Sande has shown that it is not possible to generate all the implicit edits from a large set of explicit linear inequality edits. He introduced modified Chernikova methods for generating vertex points in the region in R^n bounded by the linear inequality restraints. By the theorem of Tanahasi and Luenberger, the solutions to the error localization problem occur on the vertex points. When a record does not satisfy edits, it necessarily lies outside the region in R^n. The minimum number of fields to impute can be determined by looking at solutions on the vertexes and choosing one of them that provides a minimal solution. Because the modified Chernikova algorithm was not sufficiently fast, Schopiu-Kratina and Kovar (1989) and Filion and Schopiu-Kratina (1993) introduced heuristics, one of which speeded up computation by a factor of 60. The fastest type of Chernikova algorithms that were developed by Statistics Canada are referred to as the *GEIS-type* algorithms. Todaro (1998) used GEIS-type algorithms when he wrote the AGGIES system in SAS. Because there is no control over how long the Chernikova algorithm will take with some records, GEIS and AGGIES (Todaro 1998, 1999) software both have timing loops that kick out a record after a user-specified amount of time is exceeded. Although GEIS-type algorithms yield error-localization solutions that guarantee that a record satisfies all edits, including balance-equation edits, GEIS does not have imputation procedures that assure that balance equations are satisfied. Todaro (1998), using methods that are superficially similar to SPEER methods (Greenberg 1984, Draper and Winkler 1997), successively determined imputation intervals for individual fields by solving a series of LP problems.

A fundamental research problem with GEIS-type Chernikova algorithms is what restraints on the sets of edit-failures associated with certain records cause the algorithms to take a very long time. Is it possible to speed up the algorithm significantly in all situations? The current algorithms in GEIS and AGGIES may not be sufficiently fast when millions of records must be processed. The current upper bound in GEIS computation for an individual record is 1 minute and in AGGIES is 5 minutes. In GEIS, more than 95 percent of the records are processed in less than 0.1 second. The way that CherryPi (De Waal 1998, Pergamentsev 1998) deals additionally with the time situation is to include an upper bound of eight on the number of explicit edits that are failed. If a record fails nine or more explicit edits, then the record should be reviewed clerically and manually corrected by an analyst. The manual corrections to edit-failing record can require a number of difficult and very time-consuming iterations until the record passes edits.

The reasoning with applications GEIS and AGGIES systems is similar in that each record that is not corrected automatically must be clerically reviewed. If the reviewed records are associated with small enterprises that have negligible effect on totals, then a large amount of clerical review may not be an efficient use of resources. In applications of the new SPEER system (Draper and Winkler 1997), records failing a large number of edits are partially changed and passed through the system again as many as four times. There are two results. The first is that less than 0.1% of the records fail edits in contrast to as many as 5% that may fail if the records go through the system only once. The second is that the minimum number of fields is not changed for the records that pass through the edit system more than once. This way of applying an edit system can be much more efficient because records associated small enterprises are reviewed much more rapidly. A research question is "Is it possible to develop hybrid loops for Chernikova-type edit systems so that at higher proportion of records satisfy edits and do not have to be entirely corrected via clerical review?"

Other methods for generating extra information that is needed for error localization have only been partially successful. Garfinkel, Kunnathur, and Liepins (1986) gave an algorithm and empirical results for discrete data in which failing implicit edits are generated for each record in the main edit program. Because of the large additional amount of computation, the methods were too slow to adopt in practice.

In the new SPEER system, Draper and Winkler (1998) generated a small subset of the implicit edits induced by combinations of ratio edits and balance equations when items are required to add to a total. Their solution is only partially acceptable because the set of fields designated for change can no longer be guaranteed to be the error-localization solution (minimum number of fields to impute) for some records. Indeed, it can no longer even assure that the solution of fields to change will yield a record that satisfies all edits. The "on-the-fly" method of computing implicit edits does not compute all implicit edits and, thus, cannot yield a proper error-localization solution. Theoretical work by Winkler (1998) and Chen (1998) –even though for discrete data-- strongly suggest that all implicit edits are always needed. Draper and Winkler deal with the difficulty of not getting the proper set of fields to change by passing the newly imputed record through the edit system a second time. If it passes, then nothing else is done. If the record still fails edits, it typically fails a much smaller subset of edits and the second edit/imputation pass yields a record that passes all edits. Although the additional subroutines quadrupled the amount of computation in comparison with the original SPEER editing system (e.g., Draper and Winkler 1998), the system is still extremely fast. The program processes 1000 records in less than 4 seconds on a 200 MHz Pentium computer.

One big advantage of the Draper-Winkler approach is that the proper intervals into which imputation must be done are straightforward to compute and guaranteed theoretically to be valid. In an empirical test with a complicated set of explicit edits, Draper and Winkler had 43 of 9769 records fail after the first edit pass and 1 fail after the second edit pass. Of the 43 records, all fail very complicated implicit edits that are connected by fields that are associated with ratio edits. To deal with such complicated implicit edits in a single-pass method would require an enormous increase in the computational complexity of the system for a small percentage of records (<0.5%). These 43 records typically fail 12 or more explicit edits and/or have 6 or more of 17 fields blank. If the records for which GEIS-type algorithms take too long are similar to the 43 records in the Draper-Winkler example, then a research problem is how to create an algorithm to identify such potentially time-consuming edits so that the main Chernikova algorithm is

bypassed. The algorithms to determine the set of failing edits are very fast. The additional computation for error localization and (possibly) imputation in GEIS, AGGIES, and the new SPEER slow the systems down.

3.2 **Adjunct methods and software**

Individuals sometimes have difficulty using a FH system because the data are not in a form that can be easily used by the system. There may be comparisons of ages of individuals within households that are difficult to easily represent in the format that feeds directly into the system. This is due to resultant data structures that are too cumbersome to use directly or require too many computational resources. The creation of additional data structure is how the situation of age edits was dealt with in a prototype production system for the U.S. Decennial Census. Because there are too many combinations of ages to store and retain for each pair of individuals whose ages are being compared in an edit, new *derived* variables are defined that simplify the use of the FH software (DISCRETE edit system). To do this it is necessary to take the original set of input variables and produce a large number of new variables representing age relationships between persons in the household. For instance, if there is an edit that specifies a parent must be 12 years older than the householder, then it is necessary to create the edit E = {person1 < person2 + 12, person2_relat = parent}. The variable person1 always refers to householder because a householder must always be present. The variable V_E associated with the edits of form E takes 2 values: 1 if the condition within the brackets holds and 2 if the condition does not. The edit E (which is referred to as *derived* or conditional) replaces an edit of the form that explicitly enumerates all the different age combinations for which person1 (householder) is 12 or less years than person2 when person2 has relationship = parent.

The new set of explicitly defined edits consists of all the original edits that did not involve age relationships among two persons and the set of derived edits. When the FH system is run, it still checks the logical consistency of the entire edit system. It generates implicit edits that are based on the entire new set of explicit edits that are then used in the main edit program. The main edit program returns error-localization solutions that consist of original variables and the derived variables. Two programs are needed. The first runs against the original file, takes edit specifications from a file created by the person running the program, and creates the new variables. The entire set of variables and edits is in the format needed by the implicit-edit generation program. After running the main edit program, an additional program is needed to convert the error-localization solution that consists of original variables and derived variables into a solution in terms of the original edits only. The solutions of the original fields-to-change problem can no longer guaranteed to be the minimal number of fields to impute.

The research problems associated with this are as follows. First, is this a theoretically valid procedure? It seems straightforward to prove that it is. Second, is there a way to assure the final solution in terms of the original variables is the minimal one? Or is it just a solution that may be close to minimal? Third, is there a way to make the pre-processor and post-processor programs more user friendly? Fourth, is it straightforward to extend this to continuous variables that have been placed in a large number of categories (i.e., made to look discrete)? The solution to the fourth problem would allow much more editing of demographic surveys that contain quantitative data such as income and expenditure information.

Why are statistical agencies not using FH systems to edit discrete and continuous data simultaneously? Sande (1979) showed how to do this. He first showed how to convert discrete data to continuous data in a way that would allow solution of the error-localization problem. He then showed how to put a combination of discrete and continuous data into a form to which Chernikova's algorithm could be applied. Pargamentsev (1998) (see also De Waal 1998) have given details of how Chernikova's algorithm can be extended to combinations of discrete and continuous data. The details make clear that a research problem is creating versions of the algorithms that are sufficiently fast.

4. CONCLUDING REMARKS

This paper provides background on Fellegi-Holt methods as they are implemented for discrete and continuous data. It describes several Fellegi-Holt systems that are currently in use throughout the world in different statistical agencies. Because all the systems have limitations, it delineates a few research problems that, if solved, would improve the use of the systems.

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. A shorter version of this report that was presented at the Work Session on Statistical Data Editing at the Economic Commission for Europe meeting in Rome, Italy is available at http://www.unece.org/stats/documents/1999.06.sde.htm.

REFERENCES

Barcaroli, G., and Venturi, M. (1997), "DAISY (Design, Analysis and Imputation System): Structure, Methodology, and First Applications," in J. Kovar and L. Granquist, (eds.) *Statistical Data Editing, Volume II*, U.N. Economic Commission for Europe, 40-51.

Chen, B.-C. (1998), "Set Covering Algorithms in Edit Generation," Statistical Research Division Report 98/06 (also in American Statistical Association, Proceedings of the 1998 Section on Survey Research Methods, to appear).

Chernikova, N.V. (1964), "Algorithm for Finding a General Formula for the Non-negative Solutions of System of Linear Equations," *USSR Computational Mathematics and Mathematical Physics*, **4**, 151-158.

Chernikova, N.V. (1965), "Algorithm for Finding a General Formula for the Non-negative Solutions of System of Linear Inequalities," *USSR Computational Mathematics and Mathematical Physics*, **5**, 228-233.

DesJardins, D. (1997), "Coursebook for Exploratory Data Analysis and Graphics," Statistical Research Division, U.S. Bureau of the Census, Washington, DC, 20233-9100, USA.

DesJardins, D. (1998), "Expanded Coursebook for Exploratory Data Analysis and Graphics" Statistical Research Division, U.S. Bureau of the Census, Washington, DC, 20233-9100, USA.

De Waal, T. (1996), "CherryPi: A computer program for automatic edit error localization," Paper presented at the UN Work Session on Statistical Data Editing, 4-7 November 1996, Voorburg, the Netherlands.

De Waal, T. (1997), "A recipe for applying CherryPi to the edit process," Paper presented at the UN Work Session on Statistical Data Editing, 14-17 October 1997, (also available at http://www.unece.org/stats/documents/1997.10.sde.htm).

De Waal, T. (1998), "Improvements in Pergamentsev's algorithm for automatic error localisation in general data," Department of Statistical Methods, Statistics Netherlands, BPA number 5772-98-RSM.

Draper, L., and Winkler, W.E., (1997b), "Balancing and Ratio Editing with the New SPEER System," Statistical Research Division Report 97/05 (also appeared in American Statistical Association, Proceedings of the 1997 Section on Survey Research Methods, pp. 582-587).

Fellegi, I. P. and Holt, D. (1976), "A Systematic Approach to Automatic Edit and Imputation," *Journal of the American Statistical Association*, **71**, 17-35.

Filion, J.-M., and Schopiu-Kratina, I. (1993), "On the Use of Chernikova's Algorithm for Error Localization," *Statistics Canada Technical Report*.

Garfinkel, R. S., Kunnathur, A. S. and Liepins, G. E., (1986), "Optimal Imputation of Erroneous Data: Categorical Data, General Edits," *Operations Research*, **34**, 744-751.

Greenberg, B. G., and Surdi, R. (1984), "A Flexible and Interactive Edit and Imputation System for Ratio Edits", SRD report RR-84/18, U.S. Bureau of the Census, Washington, D.C., USA.

Kovar, J.G., MacMillan, J.H. and Whitridge, P. (1991), "Overview and Strategy for the Generalized Edit and Imputation System", Statistics Canada, Methodology Branch Working Paper BSMD 88-007E(updated in 1991).

Kovar, J.G., and Winkler, W.E., (1996), "Editing Economic Data", *American Statistical Association,*

Proceedings of the Section on Survey Research Methods, 81-87.
Little, R. A., and Rubin, D. B., (1987), *Statistical Analysis with Missing Data*, John Wiley: New York.
Nemhauser, G. L. and Wolsey, L. A., (1988), *Integer and Combinatorial Optimization*, John Wiley: New York.
Pergamentsev, S. Y. (1998), "Automatic Statistical Data Correction," Department of Statistical Methods, Statistics Netherlands, Report dated June 1998.
Pritzker, L., Ogus, J., and Hansen, M. H. (1965), "Computer Editing Methods--Some Applications and Results," *Bulletin of the International Statistical Institute, Proceedings of the 35th Session*, Belgrade, 395-417.
Rubin, D.S. (1975), "Vertex Generation in Cardinality Constrained Linear Programs," *Operations Research*, **23**, 555-565.
Sande, G. (1979), "Numerical Edit and Imputation," Proceedings of the 42nd Session of the International Statistical Institute, Manila, Philippines.
Schiopu-Kratina, I. And Kovar, J.G. (1989), "Use of Chernikova's Algorithm in the Generalized Edit and Imputation System," Statistics Canada, Methodology Branch Working Paper BSMD 89-001E.
Thompson, K. J. and Sigman, R. S. (1996), "Statistical Methods for Developing Ratio Edit Tolerances for Economic Censuses", *American Statistical Association, Proceedings of the Section on Survey Research Methods*, 166-171.
Todaro, T. A. (1997), "Adapting the SPEER Edit System to Edit Hog Data in the National Agricultural Statistics Service's Quarterly Agriculture Surveys", *American Statistical Association, Proceedings of the Section on Survey Research Methods*, 570-574.
Todaro, T. A. (1998), "Evaluation of the AGGIES Automated Edit and Imputation System**,** National Agricultural Statistics Service, U.S. Department of Agriculture, Washington, D.C. 20250-2000, December 1998, NASS Research Report, Report No. RD-98-XX.
Todaro, T. A. (1999), "Overview and Evaluation of the AGGIES Automated Edit and Imputation System," Room paper presented at the Conference of European Statisticians, 2-4 June, 1999, Rome, Italy.
Winkler, W.E. (1997a), "DISCRETE 97," Undocumented computer system, Statistical Research Division, U.S. Bureau of the Census.
Winkler, W.E. (1997b), "Edit/Imputation System for the Decennial Census," Conference on Censuses of Housing and Population, November 1997, Washington, DC, USA.
Winkler, W.E. (1998), "Set-Covering and Editing Discrete Data," Statistical Research Division Report 98/01 (also appeared in American Statistical Association, Proceedings of the 1997 Section on Survey Research, pp. 564-569).
Winkler, W. E., and Draper, L. R. (1997), "The SPEER Edit System", in *Statistical Data Editing, Volume 2*, U.N. Statistical Commission and Economic Commission for Europe, Geneva, Switzerland, 51-55.
Winkler, W. E., and Draper, L. R. (1998), "New SPEER Edit System", computer system and unpublished documentation, Statistical Research Division, U.S. Bureau of the Census, Washington, D.C., USA.
Winkler, W. E., and Petkunas, T. (1997), "The DISCRETE Edit System," in J. Kovar and L. Granquist, (eds.) *Statistical Data Editing, Volume II*, U.N. Economic Commission for Europe, 56-62.