Extending the Fellegi-Holt Model of Statistical Data Editing

William E. Winkler* and Bor-Chung Chen 1/
Bureau of the Census, Washington, DC 20233-9100
william.e.winkler@census.gov (draft 010925)

ABSTRACT
This paper provides extensions to the theory and the computational aspects of the Fellegi-Holt Model of Editing (JASA 1976). If implicit edits can be generated prior to editing, then error localization (finding the minimum number of fields to impute) can be quite rapid. In some situations, not all of the implicit edits can be generated because of the great number (> 10^30) of distinct edit patterns. The ideas in this paper are intended to determine more rapidly the approximate minimal number of fields to change in situations where not all implicit edits can be generated prior to editing. As a special case, the formal validity of Bankier's Nearest-Neighbour Imputation Method (NIM) is demonstrated.

Keywords: set-covering; integer programming; error localization

1. INTRODUCTION

Statistical data editing (SDE) are those methods that can be used to edit (i.e., clean-up) and impute (fill-in) missing or contradictory data. The 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.

Fellegi and Holt (1976, hereafter FH) provided a seminal model for SDE. Their methods have the virtues that, in one pass through the data, an edit-failing record can be assured to satisfy all edits and that the logical consistency of the entire set of edits can be checked prior to the receipt of data. The implementations of the system have had additional advantages over traditional if-then-else rule edit systems because edits reside in easily modified tables and computer code needs no modification. FH had three goals that we paraphrase:

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.

Fellegi and Holt were the first to demonstrate precisely what information was needed for correcting a record. By *correcting*, we mean changing (or filling in) values of fields so that a record satisfies all of the edits. Prior to FH, individuals were unable to account for edits that did not fail with a edit-failing record and that would fail after values in fields were changed so that the initially failing edits would no longer fail. In addition to (explicit) edits that are originally defined, FH showed that precise knowledge of implicit edits was needed. *Implicit edits* are those that can be logically derived from explicit edits. FH (Theorem 1) proved that implicit edits are needed for solving the problem of goal 1. Goal 1 is referred as the *error localization* (*EL*) problem. FH provided an inductive, existence-type proof to their Theorem 1. Their solution, however, did not deal with many of the practical computational aspects of the problem that, in the case of discrete data, were considered by Garfinkel, Kunnathur, and Liepins (1986, hereafter GKL), Winkler (1997), and Chen (1998). Because the error localization problem is NP-complete (GKL), reducing computation is the most important aspect in implementing a FH-based edit system.

The main purpose of this paper is to provide a method for EL when most, but not all, implicit edits are generated prior to editing. The algorithms are much faster than the direct integer programming methods for EL that do not use implicit edits that have been computed a priori. The speed increase is because the direct integer programming methods implicitly generate implicit edits during EL. Many implicit edits are repeatedly computed. To demonstrate our results, we build on ideas that are in or can be deduced from FH, GKL, Winkler (1997) and Chen (1998). Each of the previous papers had technical lemmas that showed how the number of computational paths could be reduced. In the much longer version of this paper, we provide a

number of technical lemmas that further reduce the number of computational paths (see http://www.census.gov/srd/www/byyear.html). The longer version gives the detail needed for understanding the main theorem of this paper and the subsequent new computational algorithms.

This paper is divided into a number of sections. The second section gives background, notation, and describes some of the limitations and strengths of previous approaches. It provides an example and several insights that serve as the motivation for the approach that we have adopted. The third section provides extensive theory and computational methods that are engendered by the theory. In the fourth section, we provide some discussion. The final section consists of concluding remarks.

2. BACKGROUND AND PREVIOUS WORK

This paper only considers FH methods as they apply to discrete data. Extensions to situations for numeric data or combinations of discrete and numeric data are straightforward.

2.1. Background on FH Theory and previous approaches

There are three error-localization methods. The first and slowest method is to use direct integer programming methods such as branch-and-bound. The method can require on the order of 10 minutes per record. The second method employs variants of a cardinality constrained Chernikova algorithm (Rubin 1975, Filion and Schopiu-Kratina 1993). The method is used in the GEIS system of Statistics Canada (in C, 1 second per record), in the CherryPi system of Statistics Netherlands (in Pascal, 2 seconds per record), and in the AGGIES system of the National Agriculture Statistical Service (SAS, more than one minute per record). Because the methods are somewhat slow, Statistics Netherlands (De Waal 2000) developed the LEO system that employs variants of the Fourier-Motzkin elimination method (in Pascal, at least 10 records per second). None of these methods or systems can deal, however, with large surveys having millions of records. For instance, with the U.S. Census of Manufactures, 4 percent of 2.5 million records (100,000) have edit records with failures. This large number of records drastically exceeds the capability of the aforementioned systems. It is not possible to clerically edit 100,000 records. Since most of the 100,000 are associated with small firms (companies), it seems reasonable to attain an FH system that could edit/impute all of the records automatically. Only the records associated with the largest companies would be clerically reviewed as an additional step of the editing. For the U.S. Decennial Census, there are 300 million records (see e.g., Chen, Winkler, and Hemmig 2000).

In this paper, we demonstrate how all of the records can be error-localized when not all of the implicit edits can be generated a priori. Chen (1998) and Winkler (1997) have shown that, if most of the implicit edits are generated prior to editing, then virtually all of the records can be properly error-localized. The methods of this paper provide a means of error-localization for a small proportion of remaining records that cannot be properly corrected due to an incomplete set of implicit edits. The methods are far faster than those based on Chernikova algorithms or Fourier-Motzkin Elimination.

2.2. Bankier's Nearest-Neighbor Imputation Method

Bankier (see e.g. 1997, 2000) introduced a successful method of using (hot-deck) donor imputation that has been used for the 1996 and 2001 Canadian Censuses and will be used for the 2006 Canadian Census. As with other donor imputation systems, the method is dependent on having a large population of high quality donors. Before describing NIM, we describe how a corresponding FH edit system that uses hot-deck imputation would work. The FH edit system would determine the minimum number of fields to change. A priori matching rules would be developed to select hot-deck donors from the set of records that satisfy all edits. If there are suitable donors, then imputed fields from the hot-deck donors will maintain the univariate distributions of the respondents. Two difficulties are associated with systems (either FH or if-then-else) that use hot-deck imputation. The first is that the matching rules may not be as good as they can be. This has been noted as a problem in the 1990 U.S. Decennial Census, the 1991 Canadian Census, and the 1991 British Census. The second is that there may not be enough suitable donors. The second problem is often not as serious in a census as it is in a smaller survey.

Bankier's NIM proceeds primarily by using donors. Each edit-failing record is matched with a large subset (say 2,000) of records that satisfy all of the edits. The ones, say 40, that have the smallest deviations in terms of the number of fields differing from the edit failing record are retained. If the same edit-failing record were considered by the Fellegi-Holt method and a donor was found that was in the 2,000 records that were searched as potential donors, then NIM could necessarily get the same donor substitution as the FH method. Even if it did not have the exact same donor, it would get a solution that was optimal in terms of the weighted, minimal number of fields to impute. NIM has an effective heuristic that allows it to deal with numeric data. Age (because of the number of values it assumes) can be considered numeric. NIM has further heuristics that work somewhat as follows. Each of the 40 edit-passing records differs from the edit-failing record on a set of fields. Fast heuristics look at subsets to determine if the record resulting from changing the values in the subset satisfy all edits. From the 40, the five best (in

terms of weighted minimal number of fields changed) are selected. One of the five is randomly selected as the donor for the hot-deck imputation.

There are two crucial advantages for a NIM system. The first is that all of the imputed records satisfy all of the edits. The second is that it finds the best matching rules automatically. From the standpoint of this paper, there is another crucial insight. By considering the set of fields in a donor record that differ from the edit-failing record, it is possible to efficiently fill-in (determine the subset of fields to change) a record. The potential value states are always two. Either leave the value in a field to its value in the original edit-failing record or change it to the value in the potential donor record. This paper gives ideas that characterize and generalize ideas from NIM. A series of technical lemmas (in the longer version) yield fast algorithms for filling in a record in situations where not all of the implicit edits can be generated. If some of implicit edits are not present, then a cover of the entering fields in the failed edits may not yield a set of fields to change that yields an edit-passing record. By a *cover*, we mean a set of fields that enter every failing edit. The lemmas give a quick way of determining additional fields that are needed for error-localization.

2.3. Notation, additional background and technical lemmas

A record $r=(y_1,...,y_n)$ in a computer file can have n fields subject to edits. For discrete edits, y takes values in $\prod Z^n$, the product space of integers. Each field y_i, i=1,...,n, corresponds to a variable that is coded. For instance, y_1 might take values 1=male and 2=female. y_2 might take values 1=single, 2=divorced, and 3=married. y_3 might correspond to age and take values 0 thru 99 or 1 thru 99. We set R_n equal to the set of values that field y_n can assume and $D = \prod R_n$. For convenience, we always assume that values in a R_n take values 1 thru k_n where the k_n integers are recodes of the k_n value states associated with field y_n. An edit E is a point set $P(E) \subseteq D$. A record r fails E is $y \in P(E)$. FH showed that an arbitrary edit E can be expressed as a union of edits E^i of a particular form. Each E^i can be expressed as $\prod E_{in}$ where E_{in} is the set of values assumed by the nth component of the points y_n in edit E^i. This form of E^i is called the *normal form*. If E_{in} is a proper subset of R_n, then field n is said to *enter* edit E^i and edit E^i is *involved* with field n. Entering fields of an edit E are those fields that are restricted by the edit E. If $\underline{E} = \{E^1, E^2, \ldots, E^m\}$, then use $P(\underline{E})$ to denote the union $\cup \{ P(E^i): E^i \in \underline{E}\}$.

If $r^0 \in P(\underline{E})$ for some set of edits $\underline{E}$, then the EL problem is to find (or possibly minimize) $\sum_{j \in J} c_j x_j$ subject to

$$y \text{ in } D - P(\underline{E})$$

and

$$x_j = 1 \text{ if } y_j = y_j^0 \qquad (2.1)$$
$$= 0 \text{ otherwise,}$$

where j = 1, …, m. The coefficient c_j is a confidence weight. In some situations, all the c_j are set to one. The vector $x = (x_1,\ldots,x_m)$ tracks the specific fields in the original record $r^0 = (y_1^0,...,y_n^0)$ that are changed.

Let r^0 in $P(\underline{E})$. Then consider the set-covering problem (SCP):

$$\text{minimize } \sum_{j \in J} c_j x_j \qquad (2.2)$$

$$\text{subject to } \sum_{j \in J} a_{ij} x_j \geq 1, \; E^i \text{ in } \underline{E}_F(r^0)$$

where

$$a_{ij} = 1 \text{ if field j enters } E^i$$
$$= 0 \text{ otherwise}$$

and $\underline{E}_F(r^0)$ is the set of edits that are failed by r^0. FH showed that the solution to (2.2) is the same as the solution to (2.1) provided the set $\underline{E}$ of edits is a complete set of edits.

In this paper, we are concerned with performing EL when the set of edit $\underline{E}$ is incomplete. Two approaches might be taken. The first is to use a heuristic to quickly determine additional fields that might be changed. Within the first approach, there are two variants. In the first variant, exemplified by NIM, the donor record yields a superset J* of the set of fields that must be changed. In the second variant, we identify a preliminary set of fields J to change (based on an incomplete set of edits $\underline{E}$. The preliminary set is also extended to a superset J* that must be changed. In each of the variants, a subset is then identified that represents the actual fields to change. In the second approach, additional implicit edits are located during the course of EL. There are two variants. In this paper, we use a method that utilizes information obtained during the edit-generation process (e.g., Winkler 1997, Chen 1998) and new ideas of this paper. In the variant due to GKL, a cutting plane algorithm (called GKL Algorithm 2) is used to identify all the failing edits during EL. Because Algorithm 2 gives significant insights into some of the information needed for reducing computation, we state it.

GKL Algorithm 2

1. Solve the SCP (2.2) and denote the solution x*.
2. Let $J = \{j \mid x_j^* = 1\}$. Fix the values of r_j for $j \notin J$ at r_{j0}, but for every j in J, let r_j assume each of the values of R_j. Test each of the $\Pi_{j \in J} |R_j|$ possible records y so defined for membership in $D - P(\underline{E})$ where $\underline{E}$ is the existing set of

explicit and implicit edits. If no such record is found, x* specifies a solution to (2.1). Otherwise, go to Step 3.

3. Find any prime cover v^0 to

$$vQ \geq 1 \qquad (2.3)$$

v binary

where $Q = (q_{ik})$ and

$q_{ik} = 1$ if E^i is failed by the kth record y of Step 2
$= 0$ otherwise.

Let $I^0 = \{i \mid v_i^0 = 1\}$.

4. Generate the implied edit E^0 given by

$$E_{0j} = \cap \{E_{ij} \; i \in I^0\}, \; j \in J \qquad (2.4)$$
$$= Rj, \quad j \notin J.$$

Let $E_F(r^0) = E_F(r^0) \cup \{E^0\}$. Go to Step 1.

GKL Algorithm 2 gives a way of finding all of the additional failing edits of the form E^0 for a record r^0. Additionally, if we take any entering field i^0 in E^0, we can iteratively expand the initial cover J to $J_1 = J \cup \{i^0\}$. At the completion of the iteration process, we have a prime cover of the failing edits for record r^0. As noted by GKL, the excessive number $\Pi_{j \in J} \mid R_j \mid$ of patterns of Step 2 typically make this procedure computationally intractable except in very small situations.

Our alternative to GKL Algorithm 2 will alleviate much of the excessive computation by making use of much more of the information available from the edit-generation process of creating an incomplete set of edits.

3. THEORY

This section contains theory and explanations that are intended to make the understanding of the computational algorithms as straightforward as possible. In the first section, we cover background on how the failing explicit and implicit edits are used to determine the exact set of fields that must be changed in an edit-failing record r. Furthermore, we show how record r is filled in. By *fill in*, we mean how new values are imputed into fields in r so that record r will no longer fail any of the explicit edits. Let the set of explicit and implicit edits $\underline{E}$ be incomplete. If r is a record that fails an implicit edit that is not in $\underline{E}$, then we indicate what can go wrong as the record is filled in. In the second section, we provide the main theorem that additionally shows, for any set R of records, how to quickly generate missing implicit edits "on-the-fly." In the fourth section, we show that the Nearest Neighbor Imputation Method (Bankier 1997, 2000) can be considered a special case of the theoretical results of this paper. Because the main theorem and the computational methods represent an extension of existing FH theory, NIM is consistent with the FH Theory.

3.1. Basic Background

FH theory gives that any prime cover J*of the entering fields of the complete set $\underline{E}$ of failing explicit and implicit edits can lead to a record r' that satisfies all edits. The record r' differs from the original record r only for the values in the fields in the cover J*. A cover is *prime* if no subset of J* is also a cover. Any prime cover J* of the failing explicit and implicit edits is said to be an *error-localization* (*EL*) solution. It satisfies the conditions (2.3) and (2.1)

Assume that an incomplete set of explicit and implicit edits exists. For the remainder of this paper, we assume that, at a minimum, the set of implicit edits must include all first-level implicit edits. There are several different ways in which error localization could be performed in the main edit program. First, if the set of incomplete edits is assumed to be nearly complete, then it may be best to take a cover J* of the failing edits in the incomplete set and fail to fill-in a record. It may be sufficient to find additional fields that need to be added to the set of fields in J*. Second, it may be best to immediately look for fields I* to add to cover J* prior to doing error localization. Third, if J* is missing a moderate number of implicit edits, then it may be better to generate additional implicit edits based on the existing set of failing edits. The generation would be an efficient hybrid of the GKL algorithm 2 that targets only one field at a time. Suitable test decks may be good for finding additional implicit edits prior to running the main edit program.

3.2. Main Theorem

Let us assume that incomplete set $\underline{E}$ contains most of the implicit edits. It is very rapid to expand $\underline{E}$ to $\underline{E}$' with implicit edits that are found via Lemma 7 (see longer version). For a given set of records R, if $\underline{E}$ is expanded to $\underline{E}$', then set $\underline{E}$' would necessarily contain all implicit edits that fail for R. The generalization of Lemma 7 to situations in which a field can assume more than two values is straightforward (see the appendix). The generalization to situations when skip patterns are present (on the survey form and in the set of edits) is not straightforward. It involves a series of technical lemmas and results that extend Winkler (1997).

We are now in the position to state the main theorem.

Theorem 1. Let $\underline{E}$ be an incomplete set of edits. Let R be a set of records that fail edits in $\underline{E}$. Let $r \in R$ be a record that fails at least one implicit edit that is not in $\underline{E}$. Let $\underline{E}_F(r)$ be the set of implicit edits in $\underline{E}$ that are failed by record r. Let $F=\{f_1, \ldots, f_n\}$ be the set of fields that

are a prime cover of $\underline{E}_F(r)$. Then it is possible to very quickly find a set of fields $F_r = \{f_{n+1}, \ldots, f_q\}$ such that $F \cup F_r$ is an EL solution for r. Furthermore, it is possible to find the implicit edits $\underline{E}_{mr}$ that are failed by r that are not in $\underline{E}_F(r)$.

Corollary 1. Let $\underline{E}$ be an incomplete set of edits. Let R be a set of records that fail edits in E. Then $\underline{E}$ can be expanded to a set of edits $\underline{E}$' that contain all of the implicit edits that fail for records in R.

In the situation when there are no skip patterns, the algorithm is straightforward.

Algorithm NS1 (no skip patterns)

1. Let $E_F(r^0)$ be the set of failing edits for record r^0. Solve the SCP (2.2) and denote the solution x*.
2. Let $J = \{j \mid x_j^* = 1\}$. Fix the values of y_j for $j \notin J$ at y_{j0}. For each $j \in J$, let $E_j^c = \cap \{F_{ji}^c, j \in J, E^i \in E_F(r^0)\}$ be the intersection of the complements of the jth entering fields. If every value f_j in E_j^c yields a newly failing explicit edit $E^{k(j)}$, then generate a new implicit edit $I^{k(j)}$ that fails for record r^0. If, for all $j \in J$, there exists no such $E^{k(j)}$, then x* is a solution to (2.1); else let $\underline{E}_1 = \{I^{k(j)}, j \in J\}$ be the set of new implicit edits. Let $E_F(r^0) = E_F(r^0) \cup \underline{E}_1$. Go to Step 1.

An alternative, much faster algorithm is

Algorithm NS2 (no skip patterns)

1. Let $E_F(r^0)$ be the set of failing edits for record r^0. Solve the SCP (2.2) and denote the solution x*.
2. Let $J = \{j \mid x_j^* = 1\}$.
3. Fix the values of y_j for $j \notin J$ at y_{j0}. For each $j \in J$, let $E_j^c = \cap \{F_{ji}^c, j \in J, E^i \in E_F(r^0)\}$ be the intersection of the complements of the jth entering fields. If every value f_j in E_j^c yields a newly failing explicit edit $E^{k(j)}$, then let d(j) be a complementary entering field in $E^{k(j)}$ that is not in J. If, for $j \in J$, there exists no such $E^{k(j)}$, then go to Step 4; else let $\underline{J}_1 = \{d(j), j \in J\}$ and $J = J \cup J_1$. Go to Step 2.
4. Find a subset of J that yields a solution to (2.1).

Algorithm NS2 is far faster than Algorithm NS1 because new implicit edits do not need to be computed at intermediate stages of the algorithm. Algorithm NS2 will generally not yield a minimal solution to (2.1). Algorithm NS1 can yield a minimal solution to (2.1) if a branch-and-bound or similarly appropriate algorithm is applied to (2.2) with the new set of implicit edits that fail for record r^0. Algorithm NS1 is far faster than GKL Algorithm 2 because it does not require enumerating all the $\Pi_{j \in J} |R_j|$ possible records y associated with the cover J and finding all existing edits that fail for them.

The heuristic algorithms of Winkler (1997) and Chen (1998) generate at least 90% of the implicit edits for large survey situations in less than 24 hours. Because a survey will not contain nearly as many records as there are in the product space or in the complete set of implicit edits, a more practical day-to-day procedure may be as follows. Generate 90% of the implicit edits using the heuristic algorithms. For a given set of survey records, generate the remaining implicit edits "on-the-fly" in the main edit program.

3.3. The Nearest-Neighbour Imputation System

Bankier (1997, 2000) introduced the Nearest-Neighbour Imputation Method (NIM). NIM has always been suitable for nearly discrete data fields such as age that are very similar to numeric fields. Bankier (2000) shows how NIM is extended to situations involving general continuous data. For convenience, we will only consider the discrete data situation. NIM has two stages. In the first, an edit-failing record r_0 is compared with a large set of edit-passing records. Using a metric that counts the number of fields that differ between the records, a group G of edit-passing records that differ on the smallest number of fields is obtained. Each record r in G differs from record r_0 on a number of fields F_r. Fast heuristics determine the approximate minimal number of fields F_r' in F_r that can be changed and still yield an edit-passing record. The final imputation for record r_0 is obtained by simple random sampling of those records that are closest in terms of the number of fields in the sets F_r'.

In potential donor records, NIM considers all of the fields that differ from the edit-failing record. Assume the initial number of differing fields is N. To determine the approximate minimum number of fields to change, NIM first determines whether single fields can be dropped. This is equivalent to determining EL solutions consisting of N-1 fields. If a solution can be found having N-1 fields, then NIM may look for solutions having N-2 fields and so on. Although NIM is the direct inspiration for the approach used in this paper, NIM methods can be considered a special case of methods used in Lemmas 3-5 (see longer version). In NIM, there is a direct computational advantage because the value of the field that must be substituted is already known. The following is another corollary to the theorem of this paper.

Corollary 2. The Nearest-Neighbor Imputation Method (NIM) is consistent with the (computational) extensions of the Fellegi-Holt model of statistical data editing as detailed in this paper.

If there is a very large set of suitable donors, then NIM will get solutions (in terms of the number of fields changed) that are as good as those obtained by FH. NIM will automatically find the best nearest-neighbor matching rules. NIM drastically reduces computation because it only considers computational paths associated with the available donors. It only considers changing the values of the fields in F_r between the existing value in record r_0 and potential donor record r.

4. DISCUSSION

As with the GKL paper, this paper primarily deals with computational extensions of the FH model. In a roundabout way, Bankier's NIM procedure provides key insights that eventually led to the computational improvements of this paper. Let a record r in R fails some of the explicit edits. Let a donor record r_1 that satisfies all of the edits. Let J* be the set of fields that differ between r and r_1. Then J* is necessarily a superset of the EL solution. The NIM method works in a top-down method. The heuristic method of this paper works in a bottom-up method in finding a superset J_1* of the EL solution. In both situations, an EL solution might be equal to the sets J* or J_1*. NIM is more straightforward because it limits the computational paths to either changing a field to the value in the donor record or leaving at the value in the original record. The heuristic method of this paper must deal with more of the possible changes in field values than those of NIM.

The main theorem of this paper provides a slower method of finding all failing implicit edits for a set of records R. It further allows finding minimal EL solutions.

5. CONCLUDING REMARKS

This paper describes theoretical and computational extensions of the Fellegi-Holt model of statistical data editing. The main application is in determining the approximate minimum number of fields to impute in situations when not all implicit edits can be generated a priori. As a special case, a theoretical justification for Bankier's Nearest-Neighbour Imputation Method is given.

1/ 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 report is released to inform interested parties of research and to encourage discussion. A much longer version of this paper with a considerable number of technical lemmas is available at http://www.census.gov/srd/www/byyear.html. Other background papers are available at http://www.unece.org/stats/documents/1997.10.sde.htm .

REFERENCES

Bankier, M., Houle, A.-M., Luc, M. and Newcombe, P. (1997), "1996 Canadian Census Demographic Variables Imputation," *American Statistical Association, Proceedings of the 1997 Section on Survey Research Methods*, 389-394.

Bankier, M. (2000), "2001 Canadian Census Minimum Change Donor Imputation Methodology," U.N. Economic Commission for Europe Work Session on Statistical Data Editing, Cardiff, UK, October 2000.

Chen, B.-C. (1998), "Set Covering Algorithms in Edit Generation," *American Statistical Association, Proc. of the Section on Statistical Computing*, 91-96.

Chen, B.-C., Winkler, W. E., and Hemmig, R. J. (2000), "Using the DISCRETE edit system for ACS Surveys," Statistical Research Division.

De Waal, T. (2000), "New Developments in Automatic Edit and Imputation at Statistics Netherlands," U.N. Economic Commission for Europe Work Session on Statistical Data Editing, Cardiff, UK, October 2000.

Fellegi, I. P. and Holt, D. (1976), "A Systematic Approach to Automatic Edit and Imputation," *Journal of fthe 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.

Nemhauser, G. L. and Wolsey, L. A., (1988), *Integer and Combinatorial Optimization*, John Wiley: New York.

Rubin, D.S. (1975), "Vertex Generation in Cardinality Constrained Linear Programs," *Operations Research*, **23**, 555-565.

Winkler, W.E. (1997), "Set-Covering and Editing Discrete Data," *American Statistical Association, Proceedings of the Section on Survey Research Methods*, 564-569.

Winkler, W. E. (1999), "The State of Statistical Data Editing," in *Statistical Data Editing*, Rome: ISTAT, pp. 169-187.