Proceedings of Statistics Canada Symposium 2001
Achieving Data Quality in a Statistical Agency: a Methodological Perspective

OPTIMIZATION TECHNIQUES FOR EDIT VALIDATION AND DATA IMPUTATION

Renato Bruni[1] , Alessandra Reale[2], Renato Torelli[2]

ABSTRACT

The paper is concerned with the problem of automatic detection and correction of inconsistent or out of range data in a general process of statistical data collecting. The proposed approach is able to deal with both qualitative and quantitative values. Our purpose is also to overcome computational limits of Fellegi-Holt approach, while maintaining its positive features. As customary, data records must respect a set of rules in order to be declared correct. By encoding the rules with linear inequalities, we develop mathematical models for the problems of interest. As a first relevant point, the set of rules itself is checked for inconsistency or redundancy, by solving a sequence of *feasibility* problems. As a second relevant point, imputation is performed by solving a sequence of *set covering* problems.

KEY WORDS: Data imputation, Mathematical model, Optimization.

1. INTRODUCTION

1.1 Description

When dealing with a large amount of collected information, a well-known relevant problem arises: perform the requested elaboration considering only correct data. Errors, or, more precisely, inconsistencies between answers or out of range answers, can be due to the original compilation of the questionnaire, or introduced during any later phase of information processing. This paper is concerned with the problem of automatic detection and correction of inconsistent or out of range data in a general process of statistical data collecting. Our attention will be focused on the problem of automatically handling of hierarchical demographic data. In order to simplify our exposition, we will be focused our attention on the problem of a Population Census.

As customary for structured information, data are organized into *records*. A record has the formal structure of a set of *fields*. Giving each field a *value*, we obtain a record instance, or, simply, a record (Ramakrishnan, Gehrke, 2000). The problem of *error detection* is generally approached by formulating a set of rules that the records must respect in order to be declared *correct*. Records not respecting such rules are declared *erroneous*. Edit rules are often written in form of *conflict* rules, which express the error condition. Nevertheless, they can easily be converted into *validity* rules, which define combination of responses considered valid and consistent. We consider rules that are verified by correct questionnaires. Given an erroneous questionnaire, the problem of *error correction* is usually tackled by changing some of its values, obtaining a *corrected questionnaire* which satisfies the above rules and is as close as possible to the (unknown) *original questionnaire* (the one we would have if we had no errors).

Many software systems, developed by Statistical Offices, deal with the problem of questionnaires correction. Our purpose is to overcome computational limits (Poirier, 1999; Winkler, 1999) of Fellegi-

[1] University of Rome “La Sapienza”, DIS, Via M. Buonarroti 12, Roma, Italy, 00185.

[2] Istat, DISS, Via A. Ravà 150, Roma, Italy, 00100.

Holt (1976) approach, while maintaining its positive features, and taking advantage of some characteristics of NIM (Bankier, 1999) and RIDA (Abbate, 1997). We construct a mathematical model of the problem, and apply state-of-the-art optimization methods developed in the Operations Research community.

2. ENCODING RULES INTO LINEAR INEQUALITIES

2.1 Data Records

In the case of a census, each record contains the answers given in one questionnaire by an entire family. A family consists in a set of individuals $I = \{1,\ldots, l\}$. We generally consider for every individual the same set of fields $F = \{f_1,\ldots, f_m\}$. Considering the above fields for every individual, we have the following kind of record structure, that we will also call *questionnaire structure* Q.

$$Q = \{f^1_1,\ldots, f^1_m, \ \ldots \ f^l_1,\ldots, f^l_m\}$$

A *questionnaire instance* q, or, simply, a questionnaire, is therefore:

$$q = \{v^1_1,\ldots, v^1_m, \ \ldots \ v^l_1,\ldots, v^l_m\}$$

Each field f^i_j, with $i = 1,\ldots, l$, $j = 1,\ldots, m$, has a *domain* D^i_j, which is the set of every possible value for that field. Since we are dealing with errors, the domains include all values that can be found on questionnaires, even the erroneous ones. Fields are usually distinguished in quantitative and qualitative ones.

2.2 Rules

A questionnaire instance q is declared correct if and only if it respects a set of rules $R = \{r_1,\ldots, r_p\}$. Each rule is usually expressed as a disjunction ($\vee$) of conditions, also called propositions (p_t) Propositions can also be negated ($\neg p_t$). Therefore, rules have the structure of clauses (i.e. a disjunction of possibly negated propositions).

$$(p_1 \vee \ldots \vee p_u \vee \neg p_{u+1} \vee \ldots \vee \neg p_v)$$

Since all rules must be respected, a conjunction ($\wedge$) of propositions is simply expressed using a set of different rules, each made of a single proposition. All other logic relations between propositions (implication, etc.) can be expressed by using only the above operators ($\vee$, $\wedge$, $\neg$). A proposition involving values of a single field is here called a *logical proposition*. A proposition involving mathematical operations between values of fields is here called *mathematical proposition*. A logical proposition is, for instance, (*age* < 14), or even (*marital status* = *married*). A mathematical propositions is, for instance: (*age - marriage duration* ≥ 14). We call *logical rules* the rules expressed only with logical propositions, *mathematical rules* the rules expressed only with mathematical propositions, and *logic-mathematical rules* the rules expressed using both type of propositions. A special case of logical rules are the ones delimitating the *feasible domain* $Ï^i_j \subseteq D^i_j$ of every field. Very often, in fact, some values of the domain are not acceptable, regardless of values of all other fields. They are called *out-of-range* values.
An example of logical rule expressing that all people declaring to be married should be at least 14 years old is: $\neg$(*marital status* = *married*) $\vee \neg$(*age* < 14). Rules delimitating the feasible domain for the field *age* are for istance: (age ≥ 0), (*age* ≤ 110).

2.3 Divisions of Domains in Subsets

We say that two values v'^{i}_{j} and v''^{i}_{j} are *equivalent* from the rules' point of view when, for every couple of questionnaires $q'_{i} = \{v_1, \ldots, v'_j, \ldots, v_m\}$ and $q'' = \{v_1, \ldots, v''_j, \ldots, v_m\}$ having all values identical except for field f_j, q' and q'' are either both correct or both erroneous.

A key point is that we can always partition each domain D^{i}_{j} into n_j subsets

$$D^{i}_{j} = S^{i}_{j1} \cup \ldots \cup S^{i}_{jnj}$$

in such a way that all values belonging to the same S^{i}_{jk} are equivalent from the logical rules' point of view. A subset for the *out-of-range* values is always present. Moreover, the value for some field can be the missing value. Such value is described as *blank*, and, depending on the field, can belong or not to the feasible domain. If the blank answer belongs to the feasible domain, the subset *blank* is also present. Otherwise, it belongs to the *out-of-range* subset.

As an example, for the integer domain D^{i}_{age}, values below 0 or above 110 are *out-of-range*. The blank answer does not belong to the feasible domain, hence belongs to the *out-of-range* subset. We have the following subsets.

$$S^{i}_{age\,1} = \{0,\ldots, 13\},\ S^{i}_{age\,2} = \{14,\ldots,17\},\ S^{i}_{age\,3} = \{18,\ldots,25\},$$
$$S^{i}_{age\,4} = \{26,\ldots,110\},\ S^{i}_{age\,5} = \{\ldots, 0\}\cup\{111, \ldots\}\cup\{blank\}$$

2.3 Variables

We define here the variables of our model. We have a set of *l*×m integer variables $z^{i}_{j} \in \{0,\ldots,U\}$, one for each domain D^{i}_{j}, a set of $l(n_1 + \ldots + n_m)$ binary variables $x^{i}_{jk} \in \{0,1\}$, one for each subset S^{i}_{jk}, and a set of $l(n_1 + \ldots + n_m)$ binary variables $\bar{x}^{i}_{jk} \in \{0,1\}$, which are the complements of the x^{i}_{jk}.

We represent value v^{i}_{j} of the questionnaire with the integer variable z^{i}_{j} by defining a mapping between values of the domain and integers numbers between 0 and an upper value U. U is such that no elements of any feasible domain maps to U.

$$\psi: D^{i}_{j} \rightarrow \{0, \ldots, U\}$$
$$v^{i}_{j} \ \# \ z^{i}_{j}$$

Qualitative domains also are mapped on the set of integer numbers by choosing an ordering. All the *out-of-range* values map to the greater number used U. The *blank* value, when belonging to the feasible domain, is encoded with the integer value immediately consecutive to the greater value of the feasible domain, but smaller than U.

We encode the membership of a value to a subset by using the binary variables x^{i}_{jk}.

$$x^{i}_{jk} = \begin{cases} 1 & \text{when } v^{i}_{j} \in S^{i}_{jk} \\ 0 & \text{when } v^{i}_{j} \notin S^{i}_{jk} \end{cases}$$

Finally, the complementary binary variables are bound the former ones by the so-called *coupling constraints*.

$$x^{i}_{jk} + \bar{x}^{i}_{jk} = 1$$

We link integer and binary variables by using a set of linear inequalities called *bridge constraints*. They impose that, when z^{i}_{j} has a value such that $v^{i}_{j} \in S^{i}_{jk}$, the corresponding x^{i}_{jk} is 1 and all others binary variables $\{x^{i}_{j1}, \ldots, x^{i}_{jk-1}, x^{i}_{jk+1}, \ldots, x^{i}_{jnj}\}$ are 0.

2.4 Encoding the Rules

Logic propositions are expressed by using the binary variables, mathematical propositions are expressed by using the integer variables. Rules involving more than one individual are expressed by using the opportune variables for the different individuals. Each logical rules having clause structure

$$(p_1 \vee \ldots \vee p_u \vee \neg p_{u+1} \vee \ldots \vee \neg p_v)$$

can be converted into the following linear inequality. We define with $x(p_t)$ and $\bar{x}(p_t)$ respectively the logical variable and the complementary logical variable corresponding to p_t, and, by selecting the set $\{p_1,\ldots, p_u\}$ of the positive propositions and the set $\{p_{u+1},\ldots, p_v\}$ of the negated propositions, we also define the corresponding incidence vectors a^π and a^ν.

$$\sum_{u=1,..,n} [a^\pi_u \, x(p_t) + a^\nu_u \, \bar{x}(p_t)] \geq 1$$

The only difference when mathematical propositions are present is that they do not correspond to binary variables but to operations between the integer variables. We limit mathematical rules to those which are linear or linearizable, since faster solution methods are available for them. In particular, we allow rules composed by a division or a multiplication of two variables. For a discussion of linearizable inequalities, see for instance (Williams, 1993). Occasionally, we need to introduce further binary variables, for instance to encode disjunctions of mathematical propositions. Note, moreover, that we developed a very precise syntax for rules. Therefore, encoding could be performed by means of an automatic procedure.
As an example, consider the following logical rule.

$$\neg(\textit{marital status} = \textit{married}) \vee \neg(\textit{age} < 14)$$

By substituting the logical variables, we have $\bar{x}^i_{\text{marital\_status married}} \vee \bar{x}^i_{\text{age } \{0,\ldots,13\}}$. This becomes the following linear inequality:

$$\bar{x}^i_{\text{marital\_status married}} + \bar{x}^i_{\text{age } \{0,\ldots,13\}} \geq 1$$

As another example, consider the following logic-mathematical rule.

$$\neg(\textit{marital status} = \textit{married}) \vee (\textit{age} - \textit{marriage duration} \geq 14)$$

By substituting the logical and integer variables, we have $\bar{x}^i_{\text{marital\_status married}} \vee (z^i_{\text{age}} - z^i_{\text{marriage\_duration}} \geq 14)$. This becomes the following linear inequality:

$$U \, \bar{x}^i_{\text{marital\_status married}} + z^i_{\text{age}} - z^i_{\text{marriage\_duration}} \geq 14$$

Altogether, from the set of rules we have a set of linear inequalities (to which we add the coupling constraints and the bridge constraints), and from the set of answers to a questionnaire we have values for the introduced variables. By construction, all and only the variable assignments corresponding to correct questionnaires satisfy all the linear inequalities, hence the linear system

$$\begin{cases} A^\pi x + A^\nu \bar{x} \geq 1 \\ A^\pi x + A^\nu \bar{x} + B z \geq b \\ x^i_{jk} + \bar{x}^i_{jk} = 1 \\ z^i_j \in \{0,\ldots,U\}, \; x^i_{jk}, \bar{x}^i_{jk} \in \{0,1\} \end{cases} \qquad (1)$$

Briefly, a questionnaire q must satisfy (1) to be correct.

3. MATHEMATICAL MODELS OF THE PROBLEMS

3.1 Validation of the Set of Rules as Feasibility

The set of rules must be free from *inconsistency* (i.e. rules must not contradict each other), and, preferably, from *redundancy* (i.e. rules must not be logically implied by other rules). In the case of real questionnaires, rules can be very numerous, since a high number of rules allows a better quality error detection.

In order to check the set of rules against inconsistency and redundancy, we study the solutions of the system of linear inequalities (1). When every possible questionnaire instance q is declared erroneous, we have the situation called *complete inconsistency* of the set of rules. When the rule inconsistency appears only for particular values of particular fields, we have the (even more insidious) situation of *partial inconsistency* of the set of rules. In a large set of rules, or in a phase of rules updating, inconsistencies may easily occur.

By encoding the set of rules into a system of linear inequalities of the form (1), complete inconsistency occurs if and only if (1) is not feasible. A partial inconsistency with respect to a subset S^i_{jk} occurs if and only if the system becomes infeasible when adding the constrain $x^i_{jk} = 1$.

Moreover, in the case of inconsistency, we are interested in restoring consistency. The approach of deleting rules corresponding to inequalities that we could not satisfy is not useful. In fact, every rule has its function, and cannot be deleted, but only modified by the human expert who writes the rules. On the contrary, the selection of the set of conflicting rules can guide the human expert in modifying them. This corresponds to selecting which part of the system causes the infeasibility, i.e. an *irreducidable infeasible subsystem* (IIS) (Amaldi, Pfetsch, Trotter, 1999). Therefore, we used a feasibility solver which, in the case of infeasible instances, is able to select an IIS.

Some rules could be logically implied by others, being therefore redundant. It would be preferable to remove them, because decreasing the number of edits while maintaining the same power of error detection can simplify the whole process and make it less error prone. The problem of logical implication (Loveland, 1978) can be formulated as a feasibility problem. A rule r_s is implied by the a set of rules R if and only if the system of linear inequalities obtained by R, together with the linear inequality obtained by the logical negation of r_s, is infeasible. It can be consequently checked if each rule r_s is redundant by testing the feasibility of the system obtained from the set of rules $\{(\{R\} \setminus r_s) \cup \neg r_s\}$ is infeasible. Redundancy of every rule can be checked by applying to each one of them the above operation.

3.2 The Imputation Problems as Set Covering

After the phase of *rules validation,* we are assured that the system (1) is feasible and has more than one solution. Detection of erroneous questionnaires q_e trivially becomes the problem of checking if the variable assignment corresponding to a questionnaire instance q satisfies (1). This operation can be performed with an extremely small computational effort.

When detected an *erroneous questionnaire* q^e, the *imputation* process consists in changing some of his values, obtaining a *corrected questionnaire* q^c which satisfies the system (1) and is as close as possible to the (unknown) *original questionnaire* q^o (the one we would have if we had no errors). Two general principles should be followed during the imputation process: to apply the minimum changes to erroneous data, and to modify as less as possible the original frequency distribution of the data (Fellegi, Holt, 1976).

Generally, a cost for changing each value of q^e is given, based on the reliability of the field. Questionnaire q^e corresponds to a variable assignment. In particular, we have a set of $l(n_1 + \ldots + n_m)$ binary values e^i_{jk} and a set of $l \times m$ integer values g^i_j.

We have a cost $c^i_{jk} \in \nabla_+$ for changing each e^i_{jk}, and a cost $H^i_j \in \nabla_+$ for changing each g^i_j.

Questionnaire q^c that we want to find corresponds to the values of the variables $(x^i_{jk}, \bar{x}^i_{jk}, z^i_j)$ at the optimal solution.

The problem of *error localization* is to find a set V of fields of minimum total cost such that q^c can be obtained from q^e by changing (only and all) the values of V. This corresponds to finding the minimum changes of erroneous data, but has little respect for the original frequency distributions.
A *donor questionnaire* q^d is a correct questionnaire which should be as close as possible to q^o. Questionnaire q^d corresponds to a variable assignment. In particular, we have a set of binary values d^i_{jk} and a set of integer values f^i_j. Donors are selected according to a distance function which can be completely specified by the user.

$$\delta: (q^e, q^d) \rightarrow \nabla_+$$

The problem of *imputation trough a donor* is to find a set W of fields of minimum total cost such that q^c can be obtained from q^e by copying from the donor q^d (only and all) the values of W. This is generally recognized to cause low alteration of the original frequency distributions, although changes caused to erroneous data may be not minimum. We are interested in solving both of the above problems.
Let us introduce $\{l(n_1 + \ldots + n_m)\}$ binary variables $y^i_{jk} \in \{0,1\}$ representing the changes we introduce in e^i_{jk}.

$$y^i_{jk} = \begin{cases} 1 & \text{if we change } e^i_{jk} \\ 0 & \text{if we keep } e^i_{jk} \end{cases}$$

Furthermore, when a donor is used, let us introduce $l \times m$ binary variables $w^i_j \in \{0,1\}$ representing the changes we introduce in g^i_j.

$$w^i_j = \begin{cases} 1 & \text{if we change } g^i_j \\ 0 & \text{if we keep } g^i_j \end{cases}$$

The minimization of the total cost of the changes can be expressed with the following objective function (where the terms $H^i_j w^i_j$ appear only in the case of imputation trough a donor).

$$\min \sum_{i=1,..,l} \sum_{j=1,..,m} \sum_{k=1,..,nj} c^i_{jk} y^i_{jk} + \sum_{i=1,..,l} \sum_{j=1,..,m} H^i_j w^i_j \qquad (2)$$

However, the constraints (1) are expressed for x^i_{jk}, $\bar{x}^i_{jk}$, z^i_j. A key issue is that there is a relation between variables in (1) and variables in (2).
In the case of error localization, this depends on the values of e^i_{jk}, as follows:

$$y^i_{jk} = \begin{cases} x^i_{jk} & \text{if } e^i_{jk} = 0 \\ 1 - x^i_{jk} & \text{if } e^i_{jk} = 1 \end{cases}$$

In fact, when $e^i_{jk} = 0$, to keep it unchanged means to put $x^i_{jk} = 0$. Since we do not change, $y^i_{jk} = 0$. On the contrary, to change it means to put $x^i_{jk} = 1$. Since we change, $y^i_{jk} = 1$. Altogether, $y^i_{jk} = x^i_{jk}$.
When, instead, $e^i_{jk} = 1$, to keep it unchanged means to put $x^i_{jk} = 1$. Since we do not change, $y^i_{jk} = 0$. On the contrary, to change it means to put $x^i_{jk} = 0$. Since we change, $y^i_{jk} = 1$. Altogether, $y^i_{jk} = 1 - x^i_{jk}$.
By using the above results, we can rewrite the objective function (2).
Therefore, the problem of error localization can be modelled as follows, where the objective function and a consistent number of constraints have a *set covering* structure (Garey, Johnson, 1976).

$$\min \sum_{i=1,..,I} \sum_{j=1,..,m} \sum_{k=1,..,nj} c^i_{jk} (1-e^i_{jk}) x^i_{jk} + \sum_{i=1,..,I} \sum_{j=1,..,m} \sum_{k=1,..,nj} c^i_{jk} e^i_{jk} \bar{x}^i_{jk}$$

Subject to the set of constraints (1).

Similarly, in the case of imputation through a donor, relation between x^i_{jk} and y^i_{jk} depends on the values of e^i_{jk} and d^i_{jk}.

$$y^i_{jk} = \begin{cases} x^i_{jk} & \text{if } e^i_{jk} = 0 \text{ and } d^i_{jk} = 1 \\ 1 - x^i_{jk} & \text{if } e^i_{jk} = 1 \text{ and } e^i_{jk} = 0 \\ 0 & \text{if } e^i_{jk} = d^i_{jk} \end{cases}$$

Note, however, that even when we do not change x^i_{jk} from e^i_{jk} to d^i_{jk}, and consequently z^i_j from g^i_j to f^i_j, we still could need to change z^i_j from g^i_j to f^i_j so that a better solution is achieved. In order to guide the choice of values for z^i_j we use the information obtained by the x^i_{jk} variables. We take for z^i_j the value of the donor when we need to make changes on the x^i_{jk} or when, even if the x^i_{jk} do not change, it is more convenient to take the donor's value.

$$z^i_j = \begin{cases} g^i_j & \text{if } \forall k \; y^i_{jk} = 0 \\ f^i_j & \text{if } \exists k : e^i_{jk} = 1, \text{ or } f^i_j \text{ yields a better solution value} \end{cases}$$

By proceeding analogously to the former case, the problem of imputation through a donor can be modelled as follows. Again, the objective function and a consistent number of constraints have a set covering structure.

$$\min \sum_{i=1,..,I} \sum_{j=1,..,m} \sum_{k=1,..,nj} c^i_{jk} (1-e^i_{jk}) d^i_{jk} x^i_{jk} + \sum_{i=1,..,I} \sum_{j=1,..,m} \sum_{k=1,..,nj} c^i_{jk} e^i_{jk} (1-d^i_{jk}) \bar{x}^i_{jk} + + \sum_{i=1,..,I} \sum_{j=1,..,m} H^i_j w^i_j$$

Subject to the set of constraints (1) and to the following additional constraints.

$$\begin{cases} z^i_j = f^i_j (w^i_j + \sum_{k=1,..,nj} y^i_{jk}/2) + g^i_j (1 - w^i_j - \sum_{k=1,..,nj} y^i_{jk}/2) \\ w^i_j \le 1 - \sum_{k=1,..,nj} y^i_{jk}/2 \\ w^i_j \in \{0,1\} \end{cases}$$

4. SOLVING THE PROBLEMS

4.1 Feasibility Problems

Given the system of linear inequalities encoding the set of rules, we initially solve the sequence of feasibility problems arising from their validation. Such problems are solved by means of an enumerative approach based on branching. Branching is essentially a strategy of *divide and conquer*. The idea is to systematically partition the linear programming feasible region into manageable subdivisions and make assessments of the integer programming problem based on these subdivisions. When moving from a region to one of its subdivisions, we add one constraint that is not satisfied by the optimal linear

programming solution over the parent region. So the linear programs corresponding to the subdivisions can be solved efficiently. See for instance Nemhauser, Wolsey (1988) for a complete explanation. Elapsed times for solving each feasibility problem is always less than 1 second.

4.2 Set Covering Problems

After detection of erroneous questionnaires, for each erroneous questionnaire q^e we solve the error localization problem. We therefore proceed by selecting a number b of donor questionnaires, by choosing the nearest ones according to our distance function. So far, for each erroneous questionnaire, b problems of imputation through a donor are solved. The imputation giving the minimum value for the objective function is chosen.
Branch-and-bound frameworks can be sometimes enhanced by adding cuts, in order to prune some branches of the search tree. Such procedures are called branch-and-cut algorithms. A *cut* is an inequality satisfied by all the feasible solutions of the integer program. The new constraints successively reduce the feasible region until an integer optimal solution is found. The cut represent a hyperplane which passes between the solution of the linear programming relaxation and the integer polytope, and cuts off a part of the relaxed polytope containing the optimal linear programming solution without excluding any feasible integer points. Branch and cut methods reveal their efficiency when instance size increases. A complete discussion can be found in Nemhauser, Wolsey (1988). Elapsed times for solving each set covering problem is again always less than 1 second.

5. CONCLUSIONS

A mathematical model of the whole imputation process allows the implementation of an automatic procedure for data imputation. Such procedure repairs the data using one donor ensuring that the marginal and joint distribution within the data are preserved. The sequence of arisen integer optimization problems can be solved to optimality in by using state-of-the-art implementation of branch-and-cut procedures. Each problem is solved to optimality in extremely short times (always less than 1 second).

REFERENCES

Abbate C. (1997) La completezza delle informazioni e l'imputazione da donatore con distanza mista minima. *Quaderni di ricerca – ISTAT n. 4/1997.*

Amaldi, E., Pfetsch, M.E., and Trotter, L., Jr. (1999), "Some structural and algorithmic properties of the maximum feasible subsystem problem", *Proceedings. of 10th Integer Programming and Combinatorial Optimization conference, Lecture Notes in Computer Science 1610,* Springer-Verlag, pp. 45-59.

Bankier, M. (1999), "Experience with the New Imputation Methodology used in the 1996 Canadian Census with Extensions for future Census" *Proceedings UN/ECE Work Session on Statistical Data Editing,* Working Paper n.24, Rome, Italy.

Fellegi, P. and D. Holt (1976), "A Systematic Approach to Automatic edit and Imputation" *Journal of the American Statistical Association*, 17, pp.35-71(353).

Garey, M. R., and D.S. Johnson (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: W.H. Freeman and Company.

Ramakrishnan, R., and J. Gehrke (2000), *Database Management Systems,* McGraw Hill.

Loveland, D. W. (1978), *Automated Theorem Proving: a Logical Basis*, North Holland.

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

Poirier, C. (1999), "A Functional Evaluation of Edit and Imputation Tools" *Proceedings UN/ECE Work Session on Statistical Data Editing*, Working Paper n.12, Rome, Italy.

Williams, H. P. (1993) *Model Building in Mathematical Programming,* Chichester: J. Wiley.

Winkler, W. E. (1999), "State of Statistical Data Editing and current Research Problems" Proceedings UN/ECE Work Session on Stat. Data Edit., Working Paper n.29, Rome, Italy.