Using Shortest-Path Algorithms to Derive Implied Ratio Edits

Brian Greenberg, Social Security Administration[1]
James T. Fagan, Bureau of the Census[1]

ABSTRACT

The SPEER (Structured Program for Economic Editing and Referrals) edit system was developed at the Census Bureau in the early 1980's for editing and imputation of data collected in the economic surveys and censuses. This edit system has been extensively used and has been incorporated into the Census Bureau's standard system for processing economic census data. For each application for which it has been adapted, the SPEER system has been customized to meet specialized data needs – as was planned in the original design. However, there have been no changes to the underlying edit methodology or to the strategy for generating implied ratio edits. In this paper we present the theory for generating implied ratio edits as a shortest path problem and introduce simple yet rigorous methods to derive implied edits. Methods discussed here have already been incorporated and successfully used in Census Bureau programs for processing economic census data.

KEY WORDS: Economic Editing, SPEER Edit System, Ratio Edits

I. Introduction

The SPEER (Structured Program for Economic Editing and Referrals) edit system was developed at the Census Bureau in the early 1980's for editing and imputation of data collected in economic surveys and censuses. The first paper on SPEER was presented by Greenberg (1981) before the name SPEER was given to this edit system. In that paper, we started by stating our goal of designing a mathematically sound edit system that could be used in a variety of surveys by customizing survey-specific routines while maintaining the integrity of general, mathematical procedures. The modules containing mathematical methods and those containing survey-specific, user supplied routines were separate, yet integrated in a single system to make the new mathematical methods broadly available while allowing for user-specified procedures customized for each application.

Over the years, that design and strategy have played out well, and the SPEER system has been used extensively to edit data for Census Bureau economic surveys and censuses. Greenberg and Petkunas (1990) describe the uses to which SPEER was put up to that time. Subsequently, the economic area of the Census Bureau has developed multipurpose, general edit and imputation software which has SPEER as the methodological core. This system makes the methodological features in SPEER available to all users while allowing customization for special purpose needs.

The generalized system was dubbed "Plain Vanilla" because it was supposed to perform all basic edit and imputation functions while avoiding complexities that often plague major edit programs. Plain Vanilla was developed just prior to the 1997 economic censuses and has been used to edit all segments of the 1997 and 2002 economic censuses. It has been expanded since its initial uses while maintaining the basic structure of SPEER at its core. For a discussion of the structure, uses and performance of Plain Vanilla; see Sigman (1997), Sigman and Wagner (1997), Wagner (2000), Thompson and Adeshiyan (2003), and Thompson, *et al* (2004).

The SPEER is well-suited for the goals of Plain Vanilla because it has general, mathematically sound, multipurpose routines that combine with survey-specific specialized rules provided which are imbedded into customized subroutines. This combination provides a blend of rigor for generalized procedures while utilizing survey-specific input; and moreover integrates the two. The underlying SPEER methodology and implementation methods are described in Greenberg (1981, 1982) and Greenberg and Petkunas (1990).

The mathematical design incorporated into SPEER is an application of the theory developed by Fellegi and Holt, (1976). Their basic idea is to start with a consistent set of user-supplied explicit edits, generate implied edits, and then they use pattern of failed edits (explicit and implied) to select a minimal set of fields to change in an edit-failing record. For arbitrary continuous linear edits, the task of generating all implied edits is quite formidable. In SPEER we work only with ratio edits, and generating implied edits for a set of explicit ratio edits is quite feasible. Since each ratio edit involves only two fields and if there are N fields on a record then the number of implied ratio edits is N(N-1).

[1] This report is released to inform interested parties of research and to encourage discussion.

In this paper we present procedures for generating implied edits from a set of explicit ratio edits by representing the ratio edits as a directed graph and using a shortest path algorithm. The process described will generate all maximal implied edits, indicate when the initial explicit edit set is not consistent, and provide useful diagnostics for subject specialists to evaluate edit criteria.

Methods discussed here have been incorporated and successfully used in Census Bureau programs for processing economic census data. This paper is extracted from a more lengthy report on this topic, which is available directly from the authors.

II. Brief Overview of SPEER

Given a family of explicit **ratio edits** provided by subject specialists, the first activity in SPEER is to generate the maximal set of implied edits. Then using these maximal implied edits, on a record-by-record basis SPEER reads continuous data records (typically response records to an economic survey instrument) and either verifies that the record passes all applicable edits or SPEER identifies a minimal set of fields to adjust so that a revised record can pass all edits. The imputation routines in SPEER then allocates values to fields not reported and alters responses in fields targeted for change, creating a revised record that is consistent; that is, passes all edits.

The SPEER system consists of four major components – Edit Generator, Edit Check, Error Localization, and Imputation. Explicit ratio edits are provided by subject specialists and implied edits, the logical consequences of explicit edits, are derived in the Edit Generator. These implied edits are used in the Edit Check routine to determine which edits are failed by an edit-failing record. The list of all edit failures is passed to the Error Localization routine, which determines fields to alter based on the pattern of edit failures (and field weights when available).

The list of fields to be imputed – because they are designated for change or were not reported – along with the unchanged fields are passed to the Imputation routine, in which fields are imputed sequentially. Values that were unchanged or imputed at an earlier stage determine the acceptance region for values yet to be imputed so that the final record with all fields having values passes all edit. After all required imputations are made, a complete record is produced in which all fields are mutually consistent and the record passes all edits.

III. Explicit Edits and Generating Implied Edits

A continuous data record can be thought of as a vector of non-negative numeric data fields, $(X_1,...,X_N)$, and a ratio edit is the requirement that the quotient of two fields lies between prescribed bounds, which are read into the system as parameters. A ratio edit between fields h and g is of the form:

$$L_{hg} \le X_h / X_g \le U_{hg}.$$

where $L_{hg} \ge 0$ and $U_{hg} \le \infty$. Given a second ratio edit between fields g and k:

$$L_{gk} \le X_g / X_k \le U_{gk},$$

a derived edit between fields X_h and X_k is the product:

$$L'_{hk} = L_{hg} L_{gk} \le X_h / X_k \le U_{hg} U_{gk} = U'_{hk}.$$

Starting with a set of explicit edits we view the edit set as a *graph* in which nodes correspond to fields and edits between fields correspond to arcs between the appropriate nodes. A graph is said to be *connected* if any two nodes can be connected by a traversal of arcs starting at one of the nodes and ending at the other. If the *edit graph* is connected, when we derive the implied edits we complete the graph – that is, any two nodes have an arc between them. If the original graph is not connected, working with the connected components we can partition the fields into subsets which can be edited independently of one another. Thus, we can assume that any two fields are linked by a path of ratio edits and the edit graph is connected.

To illustrate this, suppose we have the following set of explicit ratio edits:

$$L_{1,2} \le X_1/X_2 \le U_{1,2}, \quad L_{2,3} \le X_2/X_3 \le U_{2,3},$$

$$L_{2,4} \le X_2/X_4 \le U_{2,4}$$

The fields and ratio edits are represented by the edit graph in Figure 1. The circles are nodes corresponding to fields and the solid lines between nodes are arcs corresponding to explicit edits. When we derive all implied edits we complete the graph, that is, we connect every pair of nodes by an edit, with the implied edits shown by the dotted lines.

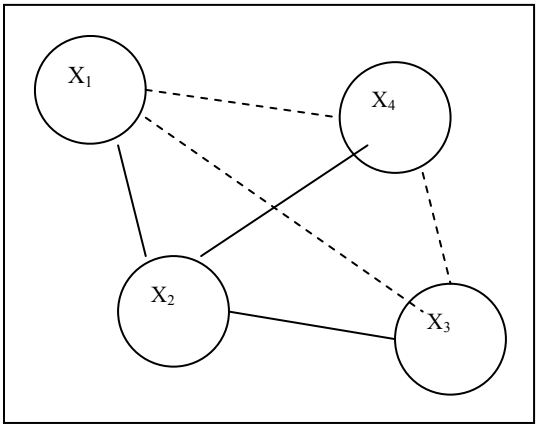

Figure 1: Edit Graph with Explicit and Implied Edits

Since the edit graph is connected, given any two integers *h* and *k* between 1 and N there is a sequence of explicit edits:

$$L_{hj_0} \le X_h / X_{j_0} \le U_{hj_0} ,$$
$$L_{j_0 j_1} \le X_{j_0} / X_{j_1} \le U_{j_0 j_1} , \ldots ,$$
$$L_{j_n k} \le X_{j_n} / X_k \le U_{j_n k}$$

whose product is a derived edit between X_h and X_k. In fact, there are typically many paths of edits from field X_h to X_k and so we get a family of derived edits involving any two fields.

For every pair of fields X_h and X_k we derive through a sequence of matrix multiplications, one at a time, a family of lower and upper limits for the ratio edit X_h/X_k, one for each path of edits from field X_h to X_k. When deriving these lower and upper limits, if at any step a new lower limit exceeds any upper limit, we halt the process and say that the explicit edit set is *not consistent* – otherwise we say the explicit edit set is *consistent*.

If an explicit edit set is not consistent, then no data record can pass all edits and the edit set is useless – and that only occurs when there was an edit misspecification. The process of generating implied edits will detect when an explicit edit set is not consistent and alerts subject specialists to an error in specifying the set of explicit edits. If an explicit edit set is not consistent, at least one explicit edit must be revised before the edits can be used.

When a connected explicit edit set is consistent we obtain the maximum lower limit and minimum upper limit for all derived edits for each ratio pair X_h/X_k over all possible derived or explicit edits. The new edit between these fields X_h/X_k defined by the optimal bounds is referred to by Fellegi-Holt as the *maximal implied edit* between fields X_h and X_k. The collection over all pairs of X_h and X_k is referred to as the set of *maximal implied edits* for an explicit edit set.

We end this section with three examples of implied edits based on a set of explicit edits. These examples show how the process of generating implied edits may relate fields not explicitly related, may tighten explicit relations between fields that were specified by subject-matter specialists, and will detect inconsistencies in the set of explicit edits.

Example 1: Suppose we have a set of records having only three fields with user-supplied edits:

$$2 \le X_1 / X_2 \le 4 \quad \text{and} \quad 1/2 \le X_2 / X_3 \le 1 ,$$

then an implied edit is

$$1 \le X_1 / X_3 \le 4 .$$

This new edit exhibits an implied relation between fields 1 and 3 based on the explicit edits.

Example 2: Suppose we again have three fields and a set of three explicit edits:

$$4 \le X_1 / X_2 \le 8 , \quad 2 \le X_2 / X_3 \le 4 , \quad \text{and}$$
$$4 \le X_1 / X_3 \le 16 .$$

Multiplying the first by the second gives the implied edit: $8 \le X_1 / X_3 \le 32$, which when combined with the third explicit edit yields the following implied edit:

$$8 \le X_1 / X_3 \le 16 .$$

Seeing how the lower bound on the implied edit differs from the lower bound of the third explicit edit, subject matter specialists must review all edits and determine what best suit their needs. If there is great faith in the first two edits, the third will have to be replaced by the implied edit. If there is great faith in the third explicit edit then one or both of the first two edits must be adjusted.

Example 3: As above, suppose we have three fields and three explicit edits:

$$8 \le X_1 / X_2 \le 16, \quad 2 \le X_2 / X_3 \le 4, \quad \text{and}$$
$$4 \le X_1 / X_3 \le 8.$$

Multiplying the first by the second and comparing to the third yields the following two relations:

$$16 \le X_1 / X_3 \quad \text{and} \quad X_1 / X_3 \le 8$$

These inequalities cannot be simultaneously satisfied by any values of the fields X_1 and X_3, and hence by no vector (X_1, X_2, X_3) can satisfy the explicit edits since they implied the relations above. The three explicit edits constitute an inconsistent edit set. Subject-matter specialists must decide which relations incorporated in the three explicit edits reflect their understanding of the data and revise the edits accordingly.

IV. Implicit Edits in the Edit and Imputation Process

In this section we show how implied edits are used in the edit and imputation process within SPEER. In addition, we describe SPEER's underlying graph structure and show how that structure is employed in editing and imputation.

When using SPEER in survey processing, the generation implied edits is performed only once after all explicit edits are specified in final form. These implied edits are used in the editing and imputation process for all survey records.

Example 4: Consider the four dimensional vector

$$(A_1,A_2,A_3,A_4) = (4,8,16,32)$$

and the three explicit ratio edits:

$$2 \le X_1/X_2 \le 4 \quad 1/4 \le X_2/X_3 \le 1 \quad 1/2 \le X_3/X_4 \le 1.$$

Note the vector fails only the first of the three ratio edits and one would expect that the field to be in error is either X_1 or X_2. Generating implied edits, we get the following maximal implied edits:

$$1/2 \le X_1/X_3 \le 4 \quad 1/8 \le X_2/X_4 \le 1 \quad 1/4 \le X_1/X_4 \le 4$$

The failed edit graph (Figure 2) shows that field X_1 fails edits with all the other fields and the values for fields X_2, X_3, and X_4 are mutually consistent; having no failed edits between any two of them and so we target field X_1 for change.

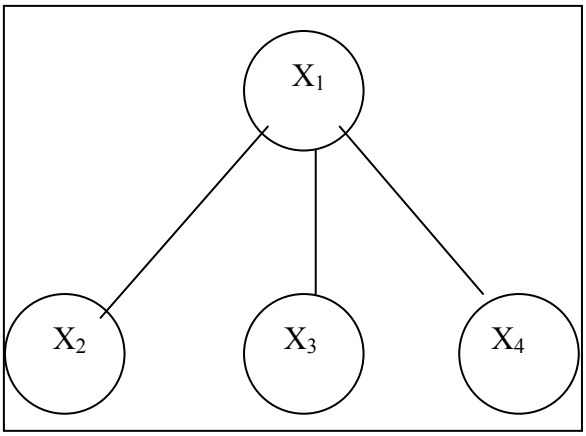

Figure 2: Failed Edit Graph

We prove below that given a consistent set of ratio edits, there exists at least one vector that passes all edits. A subset of fields on a record are said to be mutually consistent with respect to a consistent edit set if they pass all edits. We show further that if a vector has one or more missing fields and the fields present are mutually consistent with respect to the maximal implied edits, then the missing fields can be assigned values so that the completed record passes all edits.

Theorem: Given a consistent edit set, there exists a vector that passes all edits.

Proof: Assume we have a consistent edit set on N fields and so we can generate maximal implied edits:

$$L_{i,j} \le X_i/X_j \le U_{i,j}$$

with $L_{i,j} \ge 0$ for all pairs i,j = 1,…N. Our goal is to show a vector $(A_1,\ldots,A_N)$ exists such that all fields are mutually consistent, that is,

$$L_{i,j} \le A_i/A_j \le U_{i,j}$$

for all pairs i,j = 1,…N. We construct such a vector by sequentially constructing its coordinates.

To construct A_1: We can select A_1 to be any positive number.

To construct A_2: Since $L_{2,1} \le U_{2,1}$, the closed interval $[A_1L_{2,1}, A_1U_{2,1}]$ is not empty and we select A_2 to be an arbitrary value in that interval, noting that

$$L_{2,1} \le A_2/A_1 \le U_{2,1}.$$

To construct A_{k+1} given A_1 through A_k (for $k+1 \le N$): Proceeding by induction, assuming

$$L_{p,q} \le A_p/A_q \le U_{p,q}$$

for all $p,q \le k$ and consider the set of nonempty closed intervals, $[A_jL_{k+1,j}, A_jU_{k+1,j}]$ for $j \le k$.

If the intersection were empty then $A_pU_{k+1,p} < A_qL_{k+1,q}$ and so $A_p/A_q \le L_{k+1,q}/U_{k+1,p}$ for some $p,q \le k$. But

$$L_{k+1,p} \le X_{k+1}/X_p \le U_{k+1,p} \text{ and}$$

$$L_{k+1,q} \le X_{k+1}/X_q \le U_{k+1,q} \text{ so}$$

$$L_{k+1,q}/U_{k+1,p} \le X_p/X_q \le U_{k+1,q}/L_{k+1,p} \text{ and,}$$

$$L_{k+1,q}/U_{k+1,p} \le L_{p,q} \text{ and so } A_p/A_q < L_{p,q}$$

since by definition $L_{p,q}$ and $U_{p,q}$ are respectively the optimal lower and upper limits for the ratio X_p/X_q. This contradicts the assumption that $L_{p,q} \le A_p/A_q$.

Thus the intersection is not empty and we arbitrary select A_{k+1} from the intersection and observe that all ratios

$$L_{p,q} \le A_p/A_q \le U_{p,q}$$

for all $p,q \le k+1$ are satisfied. We continue the process up to $k=N-1$.

Corollary: Given a consistent edit set and a mutually consistent subset of fields, the missing fields can be assigned values so that the complete record passes all edits.

Proof: By reordering fields, we can assume that field values A_j are mutually consistent for $j=1,\ldots,k$ and the remaining fields must be assigned values. Following along the lines of the proof above, we sequentially impute fields A_j for $j=k+1, \ldots, N$ to obtain a complete consistent record.

When we assign a value to A_{k+1} to be consistent with all mutually consistent variables already on the record, the selected new value must line in the intersection of all closed intervals as indicated above. That intersection is referred to the feasible region for variable A_{k+1}.

Example 4 revisited: Since the values in fields $X_2=8$, $X_3=16$, and $X_4=32$ are mutually consistent, we can assign a value to field X_1 so that the entire record is consistent. Using all maximal implied edits that involve X_1, we get the following linear inequalities:

$$2X_2 \le X_1 \le 4X_2 \quad 1/2\, X_3 \le X_1 \le 4X_3 \quad 1/4\, X_4 \le X_1 \le 4X_4$$

and so we see that:

$$16 \le X_1 \le 32, \; 8 \le X_1 < 64, \text{ and } 8 \le X_1 \le 128$$

so: $$16 \le X_1 \le 32$$

is the feasible region for field X_1. Any value in that range will complete the record to create a record that fails no edits, and selecting $X_1=20$, yields the consistent record (20,8,16,32).

V. Edit Generation as a Shortest Path Problem

When using SPEER one starts with the explicit edits and derives the maximal implied edits. There may be several implied edits involving any two fields and from these we select the optimal limits. In earlier versions of SPEER, edit generation was performed as a sequence of matrix multiplications performed by an algorithm developed especially for that purpose. In this paper, we present a method for deriving the implied edits by capitalizing on the structure of ratio edits as a directed graph and then solving a shortest path problem.

One needs only to consider upper bounds when discussing ratio edits as each ratio edit

$$L_{hk} \le X_h / X_k \le U_{hk}$$

can be rewritten as two edits employing only upper bounds:

$$X_h / X_k \le U_{hk} \quad \text{and} \quad X_k / X_h \le U_{kh} (= 1/L_{hk}).$$

This rewriting allows us to represent edits as arcs in a directed graph and finding the optimal implied edits as shortest paths between appropriate nodes. Let $1/0 = \infty$, which in a computer program is represented by a very large number. Note that for a consistent edit set:

$$1 \le U_{hk} / L_{hk} = U_{hk}U_{kh},$$

so an edit set is not consistent when there are explicit or implied edits with $U_{hk}U_{kh} < 1$.

Starting with a few standard definitions, we use Ahuja, *et al*, (1993) as a general reference for directed graphs, shortest paths, and related material. A *directed graph* consists of a set of *nodes* and *directed arcs* between a subset of the nodes with each arc having a positive *length*. We say a directed graph is *complete* if there is a directed arc from each node to every other node. A *path* from node h and node k is a traversal of arcs, starting from node h and terminating at node k, only moving in the positive direct along each arc. A *cycle* is path which ends at the same node at which it begins. The *length* of a path is the sum of the lengths on all arcs in the path and the *shortest path* from node h to node k is the path whose sum of arc lengths is the least. Efficient algorithms are available to find shortest paths in directed graphs, and we couch the task of finding implied edits as a shortest path problem and employ standard methods to solve it.

We establish the relation between ratio edits on a continuous data set with N fields and directed graphs by constructing a directed graph on N nodes each corresponding to a field, an arc between nodes V_h and V_k corresponds to the upper limit of the ratio edit between fields X_h and X_k, and the length of this arc is $C_{hk}=\text{Log}_2 U_{hk}$.

In general, for an explicit edit set on N continuous variables, we construct a complete directed graph on N nodes in which the length of the arc from V_h to V_k has length $C_{hk}= \text{Log}_2 U_{hk}$ when there is an explicit of the form $X_h / X_k \le U_{hk}$ and the length is $C_{hk}= \infty$ otherwise.

Conversely, given a complete directed graph on N notes with length C_{hk} on the arc from node V_h to V_k we establish a family of explicit edits where an edit between X_h and X_k has upper limit:

$$X_h / X_k \le 2^{C_{hk}} = U_{hk} .$$

There is a one-to-one relationship between arcs in a directed graph with N nodes and explicit ratio edits on N continuous variables. Moreover, there is a one-to-one correspondence between implied ratio edits involving two fields and the length of paths between their corresponding nodes. The process for generating implied edits is multiplicative while the process for finding shortest paths in a directive graph is additive and the relation is as follows.

Given any two integers *h* and *k* between 1 and N there is a sequence of explicit edits:

$$L_{hj_0} \le X_h / X_{j_0} \le U_{hj_0} ,$$
$$L_{j_0 j_1} \le X_{j_0} / X_{j_1} \le U_{j_0 j_1} , \ldots ,$$
$$L_{j_n k} \le X_{j_n} / X_k \le U_{j_n k}$$

whose product is an implied edit between h and k. In fact, there are typically many paths of edits from field X_h to X_k and so we get a family of implied edits. For the sequence above, the implied upper limit between fields X_h and X_k is:

$$X_h / X_k \le U_{hj_0} U_{j_o j_1} \ldots U_{j_n k}$$

In the directed edit graph, we have nodes corresponding to the fields $X_h, X_{j_0}, \ldots, X_{j_n}, X_k$ and arcs joining the respective fields whose lengths correspond to the upper limits of edits:

$$Log_2 U_{hj_0}, Log_2 U_{j_o j_1}, \ldots, Log_2 U_{j_n k}$$

The length of the path from the node corresponding to X_h to the node corresponding to X_k has length:

$$Log_2 U_{hj_0} + Log_2 U_{j_o j_1} + \ldots + Log_2 U_{j_n k}$$

which is equal to $Log_2 (U_{hj_0} U_{j_o j_1} \ldots U_{j_n k})$.

That length corresponds to the upper limit of the edit between X_h and X_k by:

$X_h / X_k \le 2^{Log_2 (U_{hj_0} U_{j_o j_1} \ldots U_{j_n k})}$ which is the same as $X_h / X_k \le U_{hj_0} U_{j_o j_1} \ldots U_{j_n k}$

That is, starting from two fields, we (1) find a path of edits between them, (2) map the fields and path into the corresponding directed graph, (3) find the length of that path, and (4) map back to the family of ratio edits and upper bounds. Doing so we obtain exactly the same implied edit we would have obtained if we performed the multiplication of upper bounds discussed earlier.

If the length of shortest path from node V_h to node V_k is equal to C_{hk}, and the shortest path from node V_k to node V_h is C_{kh}, then the maximal implied edit between fields X_h and X_k is:

$$2^{-C_{hk}} \le X_h / X_k \le 2^{C_{hk}} .$$

We can apply Floyd's shortest path algorithm for a directed graph to obtain shortest paths in this directed graph between each ordered pair of nodes. The shortest path between an ordered pair of nodes in this directed graph will convert back to yield the implied edit between the respective fields. For evey pair of nodes, Floyd's algorithm proceeds one path at a time to find all possible paths between them, replacing the length of the arc between these nodes by successively smaller path lengths. If at any junction, the algorithm detects a negative cycle, the algorithm aborts with the message that there is no shortest path. In the absence of a negative cycle between any two nodes, Floyd's algorithm terminates at an optimal solution and the shortest path between any two nodes does exist and is obtained. When there is a negative cycle, there is no lower bound to the length of any path.

Since the length of a cycle from node V_h back to itself through node V_k is equal to

$$\text{Log}_2U_{hk}+\text{Log}_2U_{kh}=\text{Log}_2U_{hk}U_{kh}\,,$$

this cycle has negative length if and only $U_{hk}U_{kh}<1$. The condition that $U_{hk}U_{kh}<1$ is exactly the condition that a ratio edit set is not consistent. Thus, we see that a ratio edit set is not consistent if and only if there is a cycle in the edit graph whose length is less than zero.

Putting this together, starting with an explicit set of ratio edits: (1) if no lower limits exceed the upper limits on an implied ratio edit, (2) then there are no negative cycles in the corresponding ratio edit graph, (3) thus every pair of nodes has a path of shortest length between them, (4) which corresponds to the maximum implied ratio edit between every pair of nodes and (5) and hence, there is a maximum implied ratio edit for every pair of fields for the initial set of explicit edits.

Thus, transforming ratio edits to directed graphs (in the manner shown above) and getting the minimal length path between nodes, and converting back to ratio edits yields the maximal implied ratio edits needed to implement the Fellegi-Holt methodology incorporated within SPEER. That is, Floyd's shortest path algorithm does provide the optimal implied ratio edits between all fields.

We will use an earlier example to illustrate the relation between ratio edits and the directed graph. In the example below, lines between nodes are arcs represent representing ratio edits. Dashed lines represent fields not connected by a ratio edit and arc lengths (alongside the arcs) are the logarithm base 2 of the appropriate bounds of explicit ratio edits. The implied arc lengths are in boxes alongside the arcs to which they apply.

Example 1 revisited: Given the explicit ratio edits and related upper bounds:

$$2 \le X_1 / X_2 \le 4 \quad \text{and} \quad 1/2 \le X_2 / X_3 \le 1\,,$$

we have the corresponding directed graph. The solid lines represent the explicit upper limits, dotted lines represent the derived upper limits, and the values in boxes are the lengths of the derived upper limits between the appropriate fields. Note that the directed graph below indicates the implied ratio edit:

$$1 \le X_1 / X_3 \le 4\,.$$

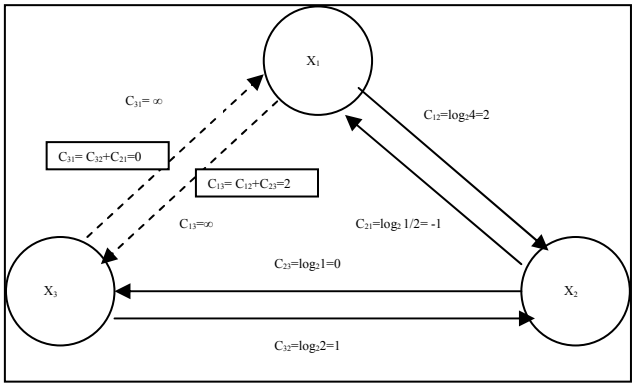

Figure 3: Directed Graph for Example 1

VI. Concluding Remarks

When reviewing algorithms incorporated in SPEER to generate maximal implied edits, we observed that the methodology there is identical to Floyd's algorithm. That is, the step-by-step multiplicative process for finding maximal implied edits in SPEER is identical to the steps for obtaining the minimal length paths under Floyd's algorithm – after converting edit bounds to their logarithms and replacing multiplication of bounds to addition of lengths. Even the stopping rule for detecting an inconsistent edit set (lower bound greater than upper bound) corresponds exactly to the stopping rule when a path length is not bounded below (presence of negative cycles).

By formulating the problem of finding maximal ratio edits as a shortest path problem one is able to couch the problem in a well-established conceptual framework. Furthermore, it allowed us to see that the algorithms in SPEER had a firm relation to

established theory. We also benefit by having a formal proof that the process in SPEER does produce optimal bounds. This formulation further highlights the intimate relation between ratio edits and directed graphs, which has been fundamental in our development of SPEER.

One benefit of explicitly using Floyd's algorithm to generate the implied edits is that algorithm can be employed to exhibit the arcs traversed in the shortest path between any two nodes. Thus one can show exactly what product of explicit edits gave rise to the implied edit and observe the path of inferences that led from the explicit to the implied edits. That is of particular value if one wants to make changes to the explicit edits based on information displayed by the implied edits. Also, if there is an inconsistency in the edit set, one can immediately observe which combination of explicit edits was responsible.

SPEER was developed in the early 1980's to provide mathematical sound methods for the editing of economic data under ratio edits integrated with subject-matter expertise. In its development, mathematical procedures and subject matter input were embedded in separate subroutines which were allowed to interact to form a flexible system that can be refined as needed. As subject-matter requirements changed, affected modules could be updated without disrupting the mathematically based procedures. As enhancements were made to theoretical methods, they could be embedded in the mathematical modules of the system while not disrupting subject-matter routines. And most importantly, neither type of change disrupts the overall flow and logic of the system as a whole. This system has seen extensive use since its development – as documented in the early papers and others cited in the references – and part of its success has been due to its flexibility, adaptability to change, and the rigor of mathematical procedures.

VII. Acknowledgements

The authors thank Paul Massell and Katherine Jenny Thompson for their careful and thoughtful review of earlier versions of this paper. Their many suggestions were extremely valuable and have been incorporated throughout. We thank Scot Dahl for information on the performance of the shortest path algorithm in generating implied edits for the economic census. We also thank Tom Petkunas for computer code used to generate implied edits in early production versions of SPEER. His code was used to compare the earlier edit generation procedures in SPEER with the shortest path algorithm.

REFERENCES

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1993) *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall.

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

Greenberg, B. 1981. "Developing and Edit System for Industry Statistics." *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, Springer-Verlag. New York, 11-16.

Greenberg, B. 1982, "Using an Edit System to Develop Editing Specification." *Proceedings of the Section on Survey Research Methods*. American Statistical Association. 366-371.

Greenberg, B and Petkunas, T. 1990. "SPEER (Structured Program for Economic Editing and Referrals)." *Proceedings of the Section on Survey Research Methods*. American Statistical Association.

Sigman, R. (1997). "Development of a 'Plain Vanilla' system for Editing Economic Census Data," paper presented at the conference of European Statisticians Work Session on Statistical Data Editing," Geneva: United Nations Economic Commission on Europe, October 1997.

Sigman, R.S. and Wagner, D. (1997). Algorithms for Adjusting Survey Data that Fail Balance Edits. *Proceedings of the Section on Survey Research Methods*. American Statistical Association.

Thompson, K, J. and Adeshiyan, S. (2003). "Data Quality Effects of Alternative Edit Parameters." *Journal of Data Science*. **1**, 1-25.

Thompson, K. J., Fagan, J., Yarbrough, B.L., and Hambric, D.L. (2004). "Using a Quadratic Programming Approach to Solve Simultaneous Ratio and Balance Edit Problems"" *Proceedings of the Section on Survey Research Methods*. American Statistical Association (to appear).

Wagner, D. (2000). "Economic Census General Editing --Plain Vanilla", *Proceedings of the International Conference on Establishment Surveys*, Alexandria, VA: American Statistical Association, pp. 561-570.