A New, Linear Programming Approach to Microdata File Merging

Richard S. Barr

J. Scott Turner

Comment

Alan J. Goldman

A New, Linear Programming Approach to Microdata File Merging

Richard S. Barr

J. Scott Turner

In analyzing economic policy, one of the most important tools currently available is the microanalytic model. The significance of this technique is underscored by its increasing use in public decision-making centers: virtually every Federal agency (and a growing number of State governments) use microanalytic models for the evaluation of policy proposals. This paper focuses on the models used extensively by the U.S. Department of the Treasury's Office of Tax Analysis (OTA) to evaluate tax revision and reform proposals for the Administration and for Congress.

One of the strengths of the microanalytic technique is its direct use of sample observations rather than aggregated data. The coupling of such a large body of detailed data with the speed and flexibility of a computer system has led to the popularity of this approach to modeling.

The validity of a model's results is tied directly to the microdata used as input. When all data used come from a single sample or subsample, quality is a function of the sampling and recording procedures. If the data from a single source are incomplete, the problem becomes more complex; multiple sources are used and

RICHARD S. BARR is with the Office of Tax Analysis, U.S. Department of the Treasury, on leave from Southern Methodist University, and J. SCOTT TURNER is with the College of Business, Oklahoma State University. Thanks are given to Darwin Klingman, University of Texas at Austin, and David Karney, Analysis, Research, and Computation, Inc., for their valuable suggestions and contributions to this paper.

files must be merged to form a composite data file. Some of the difficulties associated with the merging process and techniques for their resolution are discussed in this paper.

Until recently, merging has been performed in either an ad hoc or a heuristic manner, but research at OTA by Turner and Robbins (1974) and Turner and Gilliam (1975) has shown that an optimal merge can be defined by the solution to a large-scale, linear programming transportation problem. This optimal merging not only minimizes information loss, but also preserves the variance and covariance structure of the original files.

Because of the unusually large nature of the network optimization problems, a new state-of-the-art solution system was designed to accommodate problems of up to 50,000 constraints and 65 million variables. The system was developed by Analysis, Research, and Computation, Inc. (ARC) of Austin, Tex. under contract to OTA and is currently run on a production basis on Treasury UNIVAC computers.

This paper describes the environment of the merge problem, the optimal merge model, and the pioneering mathematical programming system devised to meet this special set of needs.

The Office of Tax Analysis

The main responsibility of OTA is evaluation of proposed tax code revisions. Since the Tax Reform Bill of 1969 there has been a tax bill every year, each requiring a great deal of detailed analysis.

In the personal tax area OTA is increasing its use of computer models to generate reports containing detailed analyses of proposed changes in the tax code. Such changes are analyzed to determine the effect they would have on the tax liability of families or individuals having certain characteristics. Variations in tax revenue are determined from analysis of individual changes occurring in a set of exhaustive and mutually exclusive classes based on such characteristics as tax return income class, family size, age of family head, race, and sex. With this breakdown it can be determined, for example, how a proposed change affects the Federal tax liability of a husband-wife filing unit (joint return) with two dependent exemptions and with an adjusted gross income between

$15,000 and $20,000. From these components, the total variation of tax revenue is determined.

The tax policy changes to be analyzed come both from the Administration via the Treasury's Assistant Secretary for Tax Policy and from the tax-related Congressional committees (Ways and Means, Senate Finance, and Joint Committee on Taxation). The process is usually iterative, with one alternative leading to another, and subject to overall constraints such as a specfic limit on the total change in revenue. As a result, the computer models may be run hundreds of times in response to a series of "what if" questions.

OTA Tax Models

Two microeconomic models in heavy use at OTA are the Federal Personal Income Tax Model and the Transfer Income Model Descriptions of these models follow.

Federal Personal Income Tax Model

The *Federal Personal Income Tax Model* is used to assess proposed tax law changes in terms of their effects on distribution of after-tax income, the efficiency with which the changes will operate in achieving their objectives, the effects the changes are likely to have on the way in which individuals compute their taxes, and the implications for the level and composition of the GNP.

For example, a proposal might be made to increase the standard deduction from 16 percent to 20 percent, increase the minimum standard deduction from $1,700 to $2,100, impose a floor on itemized medical deductions equal to 5 percent of adjusted gross income, and eliminate gasoline taxes as an allowable deduction. Because of interactions among variables, the combined effect of these changes is quite different from the sum of the isolated effects. For example, many taxpayers would switch from itemization to the standard deduction.

Transfer Income Model (TRIM)

The *Transfer Income Model* (TRIM) is an enormous and complex microdata model used by almost every Federal department for analysis of transfer income programs. It generates total budget requirements and detailed distributional effects of new transfer programs or changes to existing programs. Moreover, the model can describe the impact of simultaneous program changes. For example, TRIM can ascertain the effect of the cost-of-living component in social security on the food stamp program's transfers.

Sources of Data

The OTA models make heavy use of two sources of microdata: the Statistics of Income file and the Current Population Survey. As microdata, these files contain complete records from reporting units (individuals or households) but, for reasons of privacy and computational efficiency, only a representative subset of the population records are included. Each record is assigned a "weight" designating the number of reporting units represented by the particular record.

The resulting microdata file is a compromise between a complete census file and fully aggregated data. Thus, sufficient detail remains to support microanalysis of the population, while partial aggregation protects individual privacy and greatly diminishes the computational burden.

Statistics of Income (SOI)

This file is generated annually by the Internal Revenue Service and consists of personal tax return data. Returns are sampled at random from 15 to 20 income strata; selection rates differ by stratum and by sources of income (e.g., business or farm).

Thus, the basic microdata record is a personal tax return with 100 to 200 recorded data items, together with a weight equal to the reciprocal of the sampling rate. The sum of all weights equals the total number of returns (e.g., 82 million in 1975). The OTA tax models make use of a subsample of 50,000 records taken from this file. Comparison of a large number of tabulations produced from this subsample, with comparable tabulations based on the full SOI, show an agreement of ±.2 percent; hence the subsample provides a very accurate representation of the SOI.

Current Population Survey (CPS)

This survey is generated monthly by the Bureau of the Census, which interviews approximately 47,000 households, representing some 64,000 potential tax returns, to obtain information on work experience, education, demographics, et cetera. Questions are asked on the individual level as well as on the family level, and questions vary each month. The primary purpose of the CPS is to estimate the unemployment rate.

Each March, an in-depth survey is made that includes some sources of income that are common to the SOI and some that are not—such as social security and workman's compensation. Because of the presence of individual and household data and the inclusion of most sources of income, such data are very useful for analysis of tax policies and Federal transfer programs.

Merging Microdata Files

A typical problem in tax policy evaluation occurs when no single available data file contains all the information needed for an analysis. For example, if the policy question is the incidence and revenue effect of including Old Age Survivors Disability Insurance (OASDI) benefits in adjusted gross income, the Personal Statistics of Income (SOI) microdata file cannot be used in its original form since OASDI benefits are not included. Census files (e.g., CPS) with OASDI benefits do not of themselves allow a complete analysis of the effect of including this benefit, since information on allowable itemizations and capital gains are not in these files.

In an attempt to resolve this problem, procedures for matching or merging two microdata files have been proposed. They fall into the general categories of *exact* matches and *statistical* matches. In an exact match, the records for identical reporting units are contained in each file and are mated, usually on the basis of a unique identifier. Statistical merges (sometimes referred to as synthetic merges) involve files whose records are taken from the same population but are not necessarily from the same reporting units. In this case, matching of records is performed on the basis of their "closeness" with respect to the attributes common to the two files.

Difficulties in Obtaining Exact Matches

Insignificant Overlapping of Samples

In the OASDI example mentioned earlier, the necessary information for analysis exists in the SOI and CPS files together. However, exact matching would be useless because an insignificantly small number of persons will appear in both files. Assuming that two files are true probability samples, each with sampling rates of 1 in 1,000, the probability that a given person in one file will also appear in the other file is .001. If both files have a size of 50,000, the expected overlap of the two files is (.001) (50,000) = 50. Thus, even if exact matching were not in violation of the confidentiality strictures, the information gain for policy purposes would be insignificant.

Unique Record Indentifiers

Another prevalent problem is the absence of unique record identifiers. As a result, even given a significant overlapping of two data files, a 100 percent mapping of identical records between files is very unlikely (using common attributes) since the data values are subject to both measurement and recording errors. For example, questions about component income sources for a previous time period generally result in different answers if the respondent tries to recall such information without accurate records. Many respondents will state different amounts of "interest received last year" if asked the question at different times.

The situation in which two samples contain identical reporting units without unique identifiers is not typical when publicly available files are used. When this problem does arise, the application of a statistical matching procedure using common attributes produces as good a mapping of records as is possible, given the quality of the recorded attributes.

Issue of Confidentiality

In many situations, preservation of confidentiality precludes exact matching. For example, suppose information records with different content on the same person exist in two files and unique identifiers such as name, address, and social security number are present. A typical merging approach is a random matching of records within a given multiple component class, such as those within the same age, race, sex, and earned income class. It should be noted that the objective of matching in this context is to preserve the statistical content of the matched file while not allowing anyone to track exact information from one file to the next. Respondents are guaranteed that information given for one file will not be used to "check up" on information given for another file.

Expense

It may also be significantly more costly to achieve an exact match than a statistical match; for example, even if unique identifiers are present, many nonresponse items and recording errors are possible. A great deal of effort can be spent handling these "exception" records that cannot be matched without obtaining additional data. Depending upon the analytic purpose of the matched file, use of a statistical merging procedure may be best.

Statistical and Constrained Merges

Matching data files with the restriction that the variance-covariance matrix of data items in each file be identical to the variance-covariance matrix of the same data items in the matched file is designated as constrained matching. Examples of constrained matching are given by Budd (1971) and by Turner and Gilliam (1975).

The simplest case for statistical constrained matching occurs when two probability samples of equal size with equal record weights are merged. In this case, for purpose of matching, all record weights can be set equal to one. The condition for constrained matching is that each record in both files is matched with one and only one record in the other file. Consider two files, A and B, both with n records:

$$x_{ij}=\begin{cases}1 & \text{if } i\text{th record in file A is matched with the } j\text{th record in file B;}\\ 0 & \text{if } i\text{th record in file A is not matched with the } j\text{th record in file B;}\end{cases} \tag{1}$$

$$\sum_{i=1}^{n} x_{ij}=1, \text{ for } j=1, 2, \ldots n; \tag{2}$$

$$\sum_{j=1}^{n} x_{ij}=1, \text{ for } i=1, 2, \ldots n. \tag{3}$$

Equality constraints (2) and (3) ensure that the condition for constrained matching is met.

The Assignment Model of a Constrained Merge

Each microdata record consisting of r items can be viewed as a point in an Euclidean r-dimensional space. It can be shown for the example above that, under certain assumptions, the permutation of the records (points) in set B that satisfies the pertinent maximum likelihood condition has the following mathematical form:

minimize

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \tag{4}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1,\ i=1, \ldots n, \tag{5}$$

and

$$\sum_{i=1}^{n} x_{ij} = 1,\ j=1, \ldots n, \tag{6}$$

where

$$x_{ij} = \begin{cases} 1, \text{ if } i\text{th record in A is matched with the } j\text{th} \\ \qquad \text{record in B,} \\ 0, \text{ otherwise;} \end{cases} \tag{7}$$

$c_{ij} = f(p_{i1}, p_{i2}, \ldots p_{ir}, q_{j1}, q_{j2}, \ldots q_{jr})$;

$p_{ik} \equiv$ value of the kth common data item in record i of file A;

$q_{jk} \equiv$ value of the kth common data item in record j of file B.

The mathematical model given by expressions (4) through (7) is the assignment model. The optimal constrained matching of records in file A with records in file B is obtained by using any one of the known assignment algorithms (see Barr, Glover, and Klingman, 1977a) to find a set of x_{ij} values that minimize expression (4) while satisfying constraints (5), (6), and (7).

In this model, the function c is a metric of interrecord dissimilarity given by the extent to which the attributes in any one record differ from the same attributes in another record. The specification of this function is dependent upon the statistical properties of the data items p_{ik} and q_{jk} and, given the distribution of corresponding items, it is uniquely determined (Kadane, 1975).

For an intuitive formulation of optimal matching of two files (A and B) of equal size and with equal weights as the assignment model, see Turner and Gilliam (1975). In their paper, the parameter c_{ij} is viewed as the "distance" between record i of file A and record j of file B. Stating the constrained merging problem as determining the set of values x_{ij} that minimize the after-match aggregate distance between the records in file A and their corresponding matched records in file B also yields the assignment problem.

The Transportation Model of a Constrained Merge

A matching situation more typical of policy analysis problems is a constrained merge of two microdata files with variable weights in both files and an unequal number of records in the files. Let a_i be the weight of the ith record in file A, and let b_j be the weight of record j in file B. Suppose that file A has m records and that file B has n records. Also suppose that the following condition holds:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \tag{8}$$

The condition for a constrained matching of file A and file B is given by:

$$\sum_{j=1}^{n} x_{ij} = a_i, \text{ for } i = 1, 2, \ldots, m, \tag{9}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \text{ for } j = 1, 2, \ldots, n, \tag{10}$$

and

$$x_{ij} \geqq 0, \text{ for all } i \text{ and } j, \tag{11}$$

where x_{ij} represents the weight assigned to the composite record formed by merging record i of file A with record j of file B, with a zero value indicating that the records are not matched. An example of constrained matching using expressions (8) through (11) is given by Budd (1971).

If c_{ij} is specified as in the example of the assignment model example given earlier, and if the objective is to minimize the aggregate after-matching distance between two files (A and B) that satisfy equation (8), then the problem becomes:

minimize

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{12}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = a_i \text{ for } i = 1, 2, \ldots m, \tag{13}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \text{ for } j = 1, 2, \ldots n, \tag{14}$$

and

$$x_{ij} \geqq 0, \text{ for all } i \text{ and } j. \tag{15}$$

Note that expressions (13), (14), and (15) are the conditions for constrained matching and that the mathematical model given by (12) through (15) is a linear program. Moreover, this problem is the classical uncapacitated transportation model. This last observation is extremely important for computational reasons, as described in the next section.

The solution to this problem identifies the records in file B that are to be merged with each record in file A. In contrast with the assignment model, this problem permits a record in one file to be split or to be matched with more than one record in the other file. But since the weight of the original record is apportioned among the otherwise identical split records, the marginal and joint distributions of each file's variables are preserved.

Unconstrained matching of two microdata files is given by applying either constraints (13) or (14) but not both. In this case the variance-covariance matrix of only one of the files is preserved in the matching process. Okner (1972) describes an example of unconstrained matching which is the model of (12), (13), and (15).

A further discussion of constrained microdata matching as the transportation model is given by Turner and Gilliam (1975). A theoretical formulation of an optimal constrained merging is given in Kadane's paper elsewhere in this volume; there it is corroborated that under certain conditions constrained matching is analytically equivalent to the transportation model.

An Optimal File Merge System

In the transportation network model given above, the number of constraints is $(m+n)$. Since each x_{ij} represents the merging of two records, there are up to (mn) problem variables in a constrained file merge. These dimensions can be extremely large, considering typical sizes of m and n and the fact that the problem is totally dense (any of the mn variables might be positive). For example, to merge the CPS and SOI files directly would involve over 110,000 constraints and 3 billion variables.

Problems of this magnitude are far beyond the capability of the best general-purpose linear programming system and, even if they were divided into a series of subproblems, solution would involve an inordinate amount of machine time.

For these reasons, a specialized approach is required. Research has shown that specialized network algorithms allow solution of problems with many more constraints and variables than is possible with general-purpose linear programming methods. This is a result of such algorithms' relatively frugal use of computer storage and the high efficiency of the processing steps.

A large-scale network solution system for the merge problem was developed by the consulting firm of Analysis, Research, and Computation, Inc. under contract to OTA. This Extended Transportation System (ETS) makes use of the results of extensive research into network solution techniques carried out over the last decade by many of the persons associated with ARC.

Network Solution Methodologies—Background

A new approach in network algorithm development began in the late 1960's and sprung from a blending of the fields of operations research and computer science. The primary notion is that solution techniques should be specifically designed to enhance their implementation on computers. In the case of network algorithms, the steps of the linear programming process may be streamlined through the use of special data structures that allow efficient representation and updating of the solution basis.

Accompanying this new breed of algorithms were extensive computational studies that investigated the effectiveness of the various approaches (Glover et al., 1974; Barr, Glover, and Klingman, 1974, 1977b). Over a dozen network codes were compared using a large number of randomly generated problems. These studies showed that of all programs tested the computer codes based on specializations of the primal simplex method were the fastest, had the lowest data storage requirements, and were the most amenable to in-core/out-of-core implementation. Each of these advantages is important from the standpoint of merge problems; the enormous size of these problems has a strong impact on the two main machine resources—time and data storage.

The ETS Solution System

As ARC and OTA began to design a network solution system for the merge problem, the hardware available was a UNIVAC 1108 with only 160,000 words of 36-bit primary storage, plus disk and drum secondary mass storage. This limited amount of memory plus the enormous size of the problem precluded even the use of an available primal-simplex network code, which stores the distance data out-of-core on secondary storage and pages them, piecewise, into primary memory (Karney and Klingman, 1976). This prohibition results from the need to maintain in primary storage a solution basis of size $(6m+6n)$ words plus the data buffers. And even when the problem size was reduced to 50,000 constraints and 65 million variables, primary storage was insufficient, and the distance data alone would have encompassed almost all available mass storage.

The result was a twofold problem: first, the major data processing task of efficiently handling the cost data and, secondly, the extension of network solution technology to a new level to handle this problem. To meet these needs, the ETS design includes the following features:

- The primal simplex transportation code with the smallest known memory requirements is used. ETS employs a modification of the SUPERT code by Barr (forthcoming), which stores the solution basis in $(4m+4n)$ locations. Special packing techniques reduce this memory requirement to $(2m+2n)$, thus allowing a 50,000 constraint basis to be maintained in 100,000 words; the remaining locations are used for the object program and the data buffers. It should be noted, however, that such packing markedly increases the computational burden associated with executing the solution steps.
- Problems with fewer than (mn) variables are generated using a sampling window that restricts consideration to a subset of the possible matches for a given record. Several heuristic schemes are employed to determine this window, and these schemes are based primarily on comparisons of dominant items in the distance function so as to consider the "most likely" matches.
- The range of distance function values is reduced to 64 categories to permit exploitation of the machine wordsize by the packing scheme described above. This is necessitated by the size of the problem's dual variables (computed from sums of the c_{ij} values) and the number of bits available for their storage. Such a reduction has been found to have no significant effect on solution quality.
- The wordsize restriction also necessitates the use of a "Phase 1/Phase 2" solution approach instead of the more efficient "Big M" method of eliminating artificial variables from the solution basis. Since the actual merge problem is totally dense, these artificial variables correspond to matching possibilities that are assumed to be legitimate. However, their associated interrecord distances are unknown and are assumed to be extremely large. Phase 1 is used to drive these variables out of solution so as to form an initial feasible basis for Phase 2 optimization.

- By capitalizing on the limited number of cost classes and ordering the c_{ij} data, two new procedures can be used to compute "percentage of optimality" figures for intermediate solutions from this primal algorithm. The objective function value associated with a given primal simplex basis is an upper bound on the optimal solution value. Hence if a similar lower bound can be determined, a conservative measure of closeness to optimality can also be calculated. Such a measure can be used to terminate the solution procedure when a given suboptimal solution is deemed to be "good enough."

Normally, a feasible solution to the dual problem must be constructed (at great computational cost) in order to arrive at a lower bound on the optimal objective function value, but the special nature of this system allows implementation of one of these new theoretical developments. A complete description of these algorithms is given in the appendix.

- All input and output is double-buffered and, as a result, the system is not I/O bound. This systems programming technique is used to permit the pricing and pivoting operations to be carried out in parallel with the paging in of distance function data.

- The pricing procedure is enhanced through the use of a "candidate list" multipricing technique for pivot selection that has been shown to drop solution time for large problems to half of that required when using the best pivot selection of earlier studies (Mulvey, 1974).

- The system is written entirely in FORTRAN to increase its maintainability and portability. Of course, the use of a higher level language is not without its cost in efficiency, since assembly language programming would allow full exploitation of a particular machine's architecture. The execution times of some mathematical programming codes have been shown to improve by 30 percent to 300 percent through the inclusion of assembly coding in critical areas alone.

- ETS also includes a complete restarting capability, a command language for execution control, and report generation options.

The result of these ETS features is a system capable of solving optimization problems that are two orders of magnitude larger than any solved previously, thus establishing a new state of the art in mathematical programming.

Recent ETS Usage

In order to assess the impact of then-Secretary William Simon's fundamental tax reform proposals, a merge of the most recent CPS and SOI files was performed in the fall of 1976. The results were used in the preparation of *Blueprints for Basic Tax Reform* published by the Treasury (1977).

The merge was broken into six subproblems and each subproblem was optimized. The ETS solution statistics for two of these runs are given in table 1. It should be noted that the solution times would be markedly reduced if data packing were not used and if key portions of the system were coded in assembly language. And, since the effect of many of the system parameters such as pivoting strategy and page size has not been researched, even these extremely fast times should not be construed as the best attainable with ETS.

Recent comparisons between a FORTRAN-language primal network code and a state-of-the-art, commercial, general linear programming system (APEX III) have shown the specialized approach to be 130 times faster (Glover, Hultz, and Klingman, 1977). Using this figure as a basis of comparison, a general-purpose mathematical programming system running on a dedicated UNIVAC 1108 would require approximately five weeks to solve problem 2 of table 1.

More recently, ETS has merged files for use in analyzing the 1977 tax rebate proposal and President Carter's current basic tax reform initiative. Also, a new series of file merges are planned for the coming year; these merges will bring together for the first time microdata from a multiplicity of sources.

Whereas separate surveys for different informational needs would cost tens of millions of dollars apiece, this optimal, con-

TABLE 1.—*ETS run statistics for two example merges*

	Problem 1	Problem 2
Number of constraints	15,660	22,421
Number of CPS records	8,627	12,489
Number of SOI records	7,033	9,932
Number of variables	2,200,000	3,100,000
Solution time [1]	194 minutes (3.2 hours)	387 minutes (6.4 hours)
Number of pivots performed	167,825	272,556
Time spent in Phase 1	40%	30%

[1] Time in central processor seconds on Univac 1108 with system written in and compiled under FORTRAN V level 11A.

strained merge technique can bring about the merging of available sources for a small fraction of that amount. And, as its use continues, the ETS merge system is proving itself to be a highly cost-effective means of providing new, high-quality data resources for the public decision-making process.

Appendix: Algorithms for Computing Percent Optimality from Intermediate Solutions

Two techniques are available in the ETS solution process to compute lower bounds on the optimal objective function value from intermediate (suboptimal) solutions. These values are then used to determine a percent of optimality for the "current" solution.

Optimality Test 1

A lower bound on the optimal solution value may be derived in Phase 1 by an ordered inspection of the problem arcs by cost (c_{ij} distance) value.

In computing this bound, a restricted problem is solved using only the zero cost arcs, thus producing a solution that maximizes flow through this set of arcs. If this solution is feasible for the restricted problem then it is optimal for the entire problem. Otherwise, the best possible solution to the problem would be one in which all nonzero-cost flow would pass through arcs whose cost is 1. An iterative application of this reasoning may be used to compute a series of lower bounds on the optimal objective function.

For example, let S denote the total problem supply and $X(0)$ denote the maximum flow through the zero cost arcs. A valid lower bound on the optimal objective function value is, then,

$$B(0)=0[X(0)]+1[S-X(0)].$$

This bound may be used to evaluate the goodness of such a solution by next solving the problem using only the zero and one cost arcs. Let the sum of the flows through these arcs be denoted as $X(0,1)$ and the associated solution's objective function value be $F(0,1)$. If this second solution is feasible for the restricted prob-

[1] Source: Analysis, Research, and Computation, Inc. (1975).

lem, a percentage of optimality can be derived by a comparison of the upper bound $F(0,1)$ with $B(0)$. If this second solution is not feasible for the restricted problem, the $F(0,1)$ equals infinity and the percent optimality will be 0 percent. An improved bound may now be obtained using the same rationale. Namely, the new best objective function value must be at least:

$$B(0,1)=0[X(0)]+1[X(0,1)-X(0)]+2[S-X(0,1)].$$

(Based on an intuitive argument, it may appear that $[F(0,1)+2(S-X(0,1))]$ is a valid lower bound. However, it is not. Several textbook authors have made similar errors on related matters. For an explanation see Charnes and Klingman, 1969.)

This bound is used in the same way as the previous bound, and new bounds are generated in the same manner if the next problem is infeasible. The final bound obtained in Phase 1 may be used in Phase 2.

There are several important characteristics of this bound. First, the calculations following the solution of each restricted problem are very easy to perform. Second, the procedure requires that the cost data be ordered and that the arcs be inspected in cost-ascending order. These time-consuming requirements, however, should lead to a good bound if the optimal solution uses lower cost range values, for instance costs from zero to seven.

Optimality Test 2

The transportation problem can be stated as:

minimize

$$z=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij} \tag{A1}$$

subject to

$$\sum_{j=1}^{n}x_{ij}=a_i,\ i=1,\ 2,\ .\ .\ .\ ,\ m, \tag{A2}$$

$$\sum_{i=1}^{m}x_{ij}=b_i,\ j=1,\ 2,\ .\ .\ .\ ,\ n, \tag{A3}$$

$$\sum_{i=1}^{m}a_i=\sum_{j=1}^{n}b_j, \tag{A4}$$

and

$$x_{ij}\geqq 0 \text{ for all } i \text{ and } j, \tag{A5}$$

where m is the number of records in file A, n is the number of records in file B, c_{ij} is the distance between record i of file A and record j of file B, a_i is the weight of record i in file A, and b_j is the weight or record j in file B.

The dual problem to the transportation problem is:

maximize

$$w=\sum_{i=1}^{m} a_i u_i+\sum_{j=1}^{n} b_j v_j \tag{A6}$$

subject to

$$u_i+v_j \geqq c_{ij} \text{ for all } i \text{ and } j \tag{A7}$$

and

$$u_i,\ v_j \text{ unrestricted in sign.} \tag{A8}$$

From duality theory we know that, for any feasible solution $\{x_{ij}\}$ of the primal problem and any feasible solution $\{u_i\}$ and $\{v_i\}$ for the dual problem, $w \leqq z$. In particular, the relation holds if $\{x_{ij}\}$ is an optimal solution to the primal problem. Therefore, the objective function value w for any feasible solution of the dual problem is a lower bound for the optimal objective function value of the primal problem. Thus, if a primal feasible solution can be used to generate a dual feasible solution, this latter solution will provide a bound on the optimal solution to the primal problem.

Hence, suppose that $\{x_{ij}\}$ is a feasible solution to the primal transportation problem. The solution algorithm being used associates with this primal feasible solution a solution $\{u_i\}$ and $\{v_j\}$ to the dual problem. However, if the primal solution is not an optimal solution, the associated dual solution will not be feasible. This means that there will be arcs that violate the dual constraint (A7).

Suppose that the arc (i,j) is dual infeasible. Then from constraint (A7),

$$u_i+v_j>c_{ij}.$$

Let

$$d_{ij}=u_i+v_j-c_{ij}$$

and

$$u_i'=u_i-d_{ij}.$$

Consider the new dual solution obtained by substituting u_i' for u_i with all other variables unchanged. Then

$$u_i'+v_j=u_i-d_{ij}+v_j=u_i+v_j-(u_i+v_j-c_{ij})=c_{ij}$$

and

$$u_i'+v_j \leqq c_{ij}.$$

Therefore, the arc (i,j) is now dual feasible with respect to the new solution. Also, for any arc (i,k)

$$u_i' + v_k = u_i + v_k - d_{ij},$$

so that

$$u_i' + v_k \leq u_i + v_k,$$

since $d_{ij} > 0$. Thus, if an arc (i,k) was dual feasible in the original dual solution, it will still be dual feasible. Also, for any arc (k,j) with $k \neq i$, the value $u_k + v_j$ and the corresponding constraint (A7) will be unaffected. Therefore the dual solution obtained by replacing u_i with u_i' will have at least one fewer dual infeasible arc, namely (i,j).

The objective function value for this new solution is:

$$w' = \sum_{\substack{k=1\\k\neq i}}^{m} a_k u_k + \sum_{j=1}^{n} b_j v_j + a_i u_i'$$

or

$$w' = \sum_{k=1}^{m} a_k u_k + \sum_{j=1}^{n} b_j v_j + a_i u_i' - a_i u_i',$$

$$w' = w - a_i (u_i - u_i').$$

This procedure can be repeated for all dual infeasible arcs until a dual feasible solution is obtained. The objective function value for this final solution is then a bound on the optimal objective function value for the primal problem.

In contrast with optimality test 1, this bound requires a great deal of processing to calculate. However, as intermediate solutions approach the optimal, the bound becomes quite strong and may lead to earlier termination than the other test. In addition, this computation will verify optimality in some situations in which the other will not.

References

Analysis, Research, and Computation, Inc. "Extended Transportation System (ETS) Programmer Technical Reference Manual," 1975. (P.O. Box 4067, Austin, TX 78765)

Barr, Richard S. "Streamlining Primal Simplex Transportation Codes" (Research report). Dallas, Tex.: Southern Methodist University, School of Business Administration, forthcoming.

Barr, Richard S., Fred Glover, and Darwin Klingman. "The Alternating Basis Algorithm for Assignment Problems." *Mathematical Programming* 13 (1977), pp. 1–13. (a)

Barr, Richard S., Fred Glover, and Darwin Klingman. "Enhancements to Spanning Tree Labelling Procedures for Network Optimization." *INFOR* (1977), in press. (b)

Barr, Richard S., Fred Glover, and Darwin Klingman. "An Improved Version of the Out-of-Kilter Method and a Comparative Study of Computer Codes." *Mathematical Programming* 7 (1974), pp. 60–86.

Budd, Edward C. "The Creation of a Microdata File for Estimating the Size Distribution of Income." *Review of Income and Wealth* 17:4 (December 1971), pp. 317–334.

Charnes, A., and Darwin Klingman. "The More for Less Paradox in Distribution Problems" (Research report). Austin, Tex.: University of Texas, Center for Cybernetic Studies, 1969.

Glover, Fred, John Hultz, and Darwin Klingman. "Improved Computer-Based Planning Techniques" (Research report CCS 283). Austin, Tex.: University of Texas, Center for Cybernetic Studies, 1977.

Glover, Fred, David Karney, Darwin Klingman, and A. Napier. "A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems." *Management Science* 20:5 (1974), pp. 793–813.

Glover, F., Darwin Klingman, and Joel Stutz. "Augmented Threaded Index Method for Network Optimization." *INFOR* 12:3 (1974), pp. 293–298.

Kadane, Joseph, "Statistical Problems of Merged Data Files" (OTA Paper 6). Washington, D.C.: U.S. Department of the Treasury, Office of Tax Analysis, 1975.

Karney, David, and Darwin Klingman. "Implementation and Computational Study on an In-Core/Out-of-Core Primal Network Code." *Operations Research* 24:6 (November–December 1976), pp. 1056–1077.

Mulvey, John. "Column Weighting Factors and Other Enhancements to the Augmented Threaded Index Method for Network Optimization." Paper presented to joint ORSA/TIMS Conference, San Juan, P.R. 1974.

Okner, Benjamin. "Constructing a New Data Base from Existing Microdata Sets: The 1966 Merge File." *Annals of Economic and Social Measurement* 1 (1972), pp. 325–342.

Turner, J. Scott, and Gary B. Gilliam. "Reducing and Merging Microdata Files" (OTA Paper 7). Washington, D.C.: U.S. Department of the Treasury, Office of Tax Analysis, 1975.

Turner, J. Scott, and Gary A. Robbins. "Microdata Set Merging Using Microdata Files" (Research report). Washington, D.C.: U.S. Department of the Treasury, Office of Tax Analysis, 1974.

U.S. Department of the Treasury. *Blueprints for Basic Tax Reform*, January 17, 1977.

COMMENT

Alan J. Goldman, National Bureau of Standards

Barr and Turner's paper explains how the problem of optimally merging microdata files leads to special linear programs ("assignment" and "transportation" problems) of extraordinary size. It goes on to describe the Extended Transportation System (ETS), an innovative family of computer programs created to solve such problems. Novel features of ETS are sketched, and several successful applications are noted.

These developments are quite exciting, both for the mathematical-optimization community and for those engaged in supporting policy analyses through predicting the impacts of alternative policy changes. The techniques are by no means limited to the area of tax analysis; as knowledge of these methods spreads, applications to file-merging problems in other fields should proliferate.

The methods *do* appear intrinsically limited to the merging of two files (as opposed to three or more). Thus it will be interesting to see how the anticipated merging of "microdata from a multiplicity of sources," mentioned near the paper's end, will be executed. The result of a sequence of pairwise merges has an undesirable dependence on the ordering of that sequence and could differ significantly from a true multimerge optimum.

The paper notes that matching of files is to be performed under the restriction that "the variance-covariance matrix of data items in each file be identical to the variance-covariance matrix of the same data items in the matched file." This phrasing is not entirely clear, seemingly requiring (the very stringent condition) that the two original files have identical variance-covariance matrices. Moreover, although it is asserted that this restriction is satisfied by the optimization models formulated in the paper, that relationship is far from self-evident; references supporting this claim, mainly unpublished OTA papers, were not supplied for use in this review. Similar remarks about clarity and evidence apply to the assertion that the optimization models can be derived from a maximum-likelihood analysis. A clearer, more self-contained treatment of this material would have been preferable, so that the reader is not left uneasy about these important points.

The "variance-covariance matrix" condition imposed on the merged file is clearly desirable; I presume omission of a corresponding condition on mean values was just an oversight. But these conditions may not provide a full theoretical basis for all the desired uses of that file. (This is not to adopt the ultra-purist attitude that analyses lacking a totally rigorous foundation are forbidden, but only to point out that an analysis should in principle be accompanied by an account of what "ideal" supporting data would be like, how the information actually at hand falls short of the ideal, and what the anticipated possible consequences are.) Discussion of this point would be useful, as would a preview of whatever efforts are planned to validate empirically the use of the merged files. These questions probably exceed the scope of the particular paper under discussion but nevertheless need to be raised.

The use of the word "network" repeatedly in the text may merit explanation. It refers to the movement of a homogeneous commodity over a transportation network that joins each of a set of origin nodes to each of a set of destination nodes. Here, x_{ij} represents the quantity shipped from the ith origin to the jth destination, and c_{ij}, the unit cost of transportation over this network arc; thus the paper's equation (12) calls for minimization of total costs. Equations (13) and (14) identify a_i as the (fixed) total supply at origin i and b_j as the (fixed) total demand at destination j; thus equation (8) expresses the balance of supply and demand. If $m=n$ and all a_i and b_j are unity, this "transportation problem" has an alternative interpretation as an "assignment problem" involving n persons and n jobs. Here, x_{ij} is 1 or 0 depending on whether person i is or is not assigned to job j, and c_{ij} is the cost of this assignment; equations (5) and (6) reflect the fact that each person is assigned to exactly one job and vice versa. Although the applications in Barr and Turner's paper refer to files and records rather than to transportation or job-assignments, the mathematical formulations carry over.

Writing for a mixed audience is, of course, a difficult business. I suspect that much of the terminology of linear-program calculations will convey little to many readers, but this could be remedied only through lengthy discussions. Terms such as "double-buffered," "restarting" (from what?), and the like might have been defined. The following are two miscellaneous notes on substantive matters.

First, has the effect of the "sampling window" on solution quality been evaluated? What tests were used, and where are they documented? Similar questions apply to the 64-value discretization of distance-function values.

Second, the termination criteria defining "solution times" in table 1 are not specified.

REPLY

Richard S. Barr and J. Scott Turner

The following discussions are being added in response to Dr. Goldman's thoughtful critique: first, a graphical depiction and further explanation of the transportation network model as it applies to the merge process; second, derivations of the item means, variances, and covariances in the constrained merge file to show their equivalence to the statistics of the original files.

Characteristics of Transportation Problems

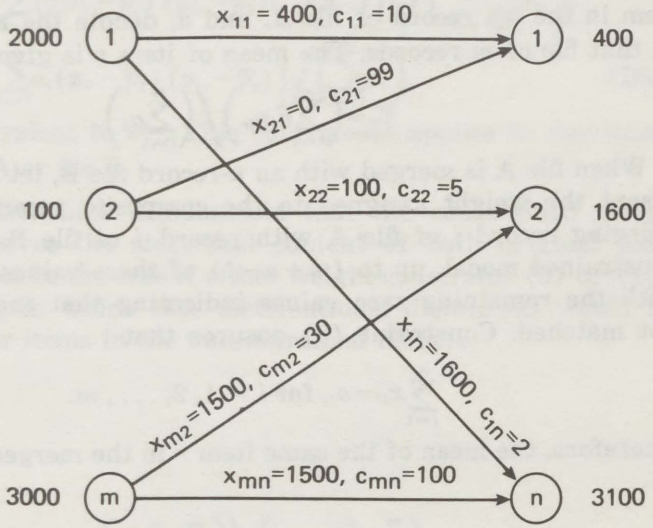
A *network* of flows is a structure that consists of a set of *nodes* and a set of *arcs* connecting some or all pairs of these nodes. When a network is presented in the form of a graph, as in figure 1, circles are used to represent nodes and lines with arrowheads to represent arcs, which are assumed to have direction.

Units of flow move between pairs of nodes across arcs, according to the arcs' given directions. Nodes that supply flow units are referred to as *origins* and those that demand flow are called *destinations.*

Each arc (i, j) corresponds to the variable x_{ij}, representing the flow from origin i to destination j, shipped at the unit cost c_{ij}. For each node there is a corresponding constraint on the amount of flow shipped out of that node (if an origin) or into the node (if a destination). The objective is to determine a pattern of shipping that meets all demand requirements and supply restrictions at a minimum overall cost.

In the merge model analogy, the nodes represent individual microdata records whose weights are given as the supply and demand values. The network arcs correspond to record matching combinations, and the associated flows and costs represent the merge record weights and distance function values, respectively. The choice of which file is to be the set of origins and which is to be the destinations has no effect on the results of the merge.

FIGURE 1.—*Example constrained merge as represented by transportation network model*


2000
1
$x_{11} = 400, c_{11} = 10$
1
400
$x_{21} = 0, c_{21} = 99$
100
2
$x_{22} = 100, c_{22} = 5$
2
1600
$x_{m2} = 1500, c_{m2} = 30$
$x_{1n} = 1600, c_{1n} = 2$
3000
m
$x_{mn} = 1500, c_{mn} = 100$
n
3100

Network Component:	Supply Values (a_i)	Origin Nodes	Arcs, with Flows and Unit Costs (x_{ij}, c_{ij})	Destination Nodes	Demand Values (b_j)
Merge Model Equivalent:	CPS Record Weights	CPS Records	Record Matches with Assigned Weights and Distances	SOI Records	SOI Record Weights

Preservation of Item Statistics in Merging

In this section we show that the means and variance-covariance matrices of items in a given file A are preserved in a file resulting from a fully constrained statistical merge with another file B. This is a consequence of including constraints for the original record weights in the merge process. This discussion does not apply to any relationships between items that were originally in different files.

Arithmetic Mean

The arithmetic mean of a data item in the merge file will retain its value from the originating file even though records may be split in the matching process. This is because the sum of the weights of any split records equals the weight of the original record.

To demonstrate this, let p_{ir} represent the value of the rth data item in the ith record of file A, and a_i denote the record weight in that file of m records. The mean of item r is given as

$$\bar{p}_r=\left(\sum_{i=1}^{m}a_i p_{ir}\right)\Big/\left(\sum_{i=1}^{m}a_i\right). \tag{B1}$$

When file A is merged with an n-record file B, let x_{ij} again represent the weight assigned to the composite record formed by merging record i of file A with record j of file B. In the fully constrained model, up to $(m+n-1)$ of these values are positive, with the remaining zero values indicating that the records are not matched. Constraint (9) ensures that

$$\sum_{j=1}^{n}x_{ij}=a_i, \text{ for } i = 1, 2, \ldots, m. \tag{B2}$$

Therefore, the mean of the same item r in the merged file is given as

$$\begin{aligned} p^*_r &=\left(\sum_{i=1}^{m}\sum_{j=1}^{n}p_{ir}x_{ij}\right)\Big/\left(\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}\right),\\ &=\left[\sum_{i=1}^{m}p_{ir}\left(\sum_{j=1}^{n}x_{ij}\right)\right]\Big/\left(\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}\right),\\ &=\left(\sum_{i=1}^{m}p_{ir}a_i\right)\Big/\left(\sum_{i=1}^{m}a_i\right), \end{aligned} \tag{B3}$$

which is equivalent to the expression for $\bar{p}_r$. This relationship holds for any item in either of the original files.

Variance-Covariance Matrices

For a similar analysis of the items' variance-covariance properties, let p_{ir} and p_{is} represent, respectively, the rth and sth data items in the ith record of file A. The following expression defines σ^2_{rs} as the variance of item r (if $r=s$) or the covariance of the two items (if $r\neq s$) in the original file:

$$\sigma^2_{rs} = \left[\sum_{i=1}^{m}a_i\,(p_{ir}-\bar{p}_r)\,(p_{is}-\bar{p}_s)\right]\Big/\left(\sum_{i=1}^{m}a_i\right). \tag{B4}$$

In a fully constrained merge file, the variances and covariances are given as

$$\sigma^{*2}_{rs} = \left\{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[x_{ij}\,(p_{ir}-p^*_r)\,(p_{is}-p^*_s)\right]\right\}\Big/\left(\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}\right). \tag{B5}$$

Since $p^*_r = \overline{p}_r$ and $p^*_s = \overline{p}_s$,

$$\begin{aligned}\sigma^{*2}_{rs} &= \left\{ \sum_{i=1}^{m} \left[(p_{ir} - \overline{p}_r)\,(p_{is} - \overline{p}_s) \left(\sum_{j=1}^{n} x_{ij} \right) \right] \right\} \Bigg/ \left(\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \right), \\ &= \left[\sum_{i=1}^{m} a_i\,(p_{ir} - \overline{p}_r)\,(p_{is} - \overline{p}_s) \right] \Bigg/ \left(\sum_{i=1}^{m} a_i \right), \qquad \text{(B6)}\end{aligned}$$

which is equivalent to σ^2_{rs}. This equivalence applies to any items in either file A or file B.

These relationships demonstrate that the constrained merge process preserves the statistical content of both original files. Such would not be the case if either weight constraint (9) or (10) were omitted, in which case distributional distortions would be introduced for items in the unconstrained file(s).