Mathematical Programming Study 15 (1981) 1–22.
North-Holland Publishing Company

MICRODATA FILE MERGING THROUGH LARGE-SCALE NETWORK TECHNOLOGY

Richard S. BARR

Edwin L. Cox School of Business, Southern Methodist University, Dallas, TX, U.S.A.

J. Scott TURNER

School of Business Administration, Oklahoma State University, Stillwater, OK, U.S.A.

Received 30 May 1979
Revised manuscript received 8 April 1980

This paper describes the results of over four years of research, development, implementation and use of a system which will optimally merge microdata files. The merge process requires solving transportation problems with up to 50 000 constraints and 60 million variables, and is now being performed on a production basis. The resultant statistical data files fuel U.S. tax policy evaluation models which are used today in the design and analysis of Federal legislation. Computational experience with this pioneering optimization software is described.

Key words: Network, Large-Scale Optimization, Microsimulation, Linear Programming, Microdata, File Merging.

1. Introduction and overview

In analyzing economic policy, one of the most important tools currently available is the microanalytic model. With this class of econometric models, calculations are performed on individual sample observations of decision units, called *microdata*, to forecast aggregate population effects of changes in public policy. The significance of this technique is underscored by its use in virtually every Federal agency and a growing number of State governments 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 need for high quality and completeness of these models' microdata is evident from the importance of their end-use applications: legislation design and public policy analysis. But for a variety of reasons, including cost and legality, data is rarely collected specifically for policy models. Instead, they inevitably rely on data accumulated as a part of program implementation (for example, I.R.S. tax forms) or from a survey commissioned for a different purpose (e.g., Census Bureau data). Therefore, the

quality of a model's data often depends on more than the sampling and recording procedures; the data from a single source may be ill-specified or incomplete. In this case, the problem becomes more complex; multiple sources are used and files are *merged* to form a composite data file.

Merging involves matching each observation record in one file with one or more records in another file. In this manner, composite records are formed which contain the data items from both original files. This paper explores some of the difficulties associated with the merging process and describes a new technique for their resolution.

Until recently, merging has been performed in either an ad hoc or a heuristic manner, but research at OTA [23, 24] 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 produces the best possible match but also preserves the complete statistical 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 and is currently run on a production basis on Treasury computer systems. 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.

In summary, public policy models often require data that is unavailable from existing sources and separate surveys would cost tens of millions of dollars apiece. The file merging process described herein is used to combine available sources for a small fraction of that cost. Thus, the objective of the optimal merging approach is the cost-effective preparation of high-quality data for input to the public decision-making process.

2. OTA tax models

The main responsibility of OTA is the evaluation of proposed tax code revisions. In the personal tax area, proposed changes are analyzed to determine the effect they would have on the tax liability of families or individuals having certain characteristics. From the analysis of a set of exhaustive and mutually exclusive classes (based on such characteristics as tax return income class, family size, age of family head, and demographics) 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 Com-

mittee on Taxation). The process is usually iterative, with one alternative leading to another, and subject to overall constraints such as a specific 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.

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

2.1. *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 $2200 to $2600, 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.

2.2. *Transfer Income Model* (TRIM)

The *Transfer Income Model* (TRIM) is an enormous and complex micro-analytic model used by almost every Federal department for analysis of transfer income programs such as welfare and social security. 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.

3. Sources of microdata

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 micro-analysis of the population, while partial aggregation protects individual privacy and greatly diminishes the computational burden.

3.1. *Statistics of Income* (SOI)

The SOI 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). For computational efficiency, 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 ± 0.2 percent; hence the subsample provides a very accurate representation of the full SOI.

3.2. *Current Population Survey* (CPS)

This survey is generated monthly by the Bureau of the Census, which interviews approximately 67 000 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 on income, such data are very useful for analysis of tax policies and Federal transfer programs.

4. 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

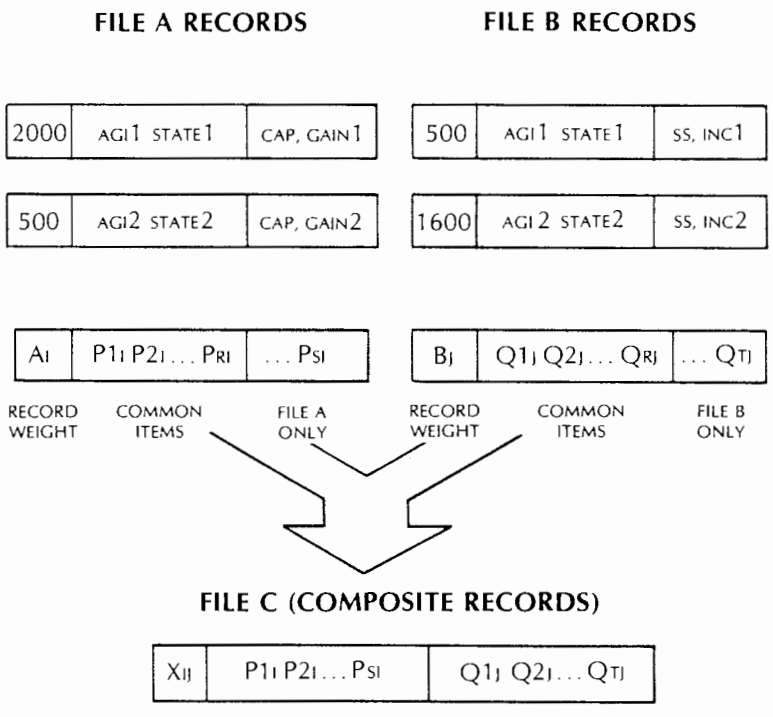
including this benefit, since information on allowable itemizations and capital gains are not in these files.

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 such as the social security number. Statistical 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, as illustrated in Fig. 1.

4.1. Difficulties in obtaining exact matches

While in many instances exact matching may be the preferable approach, in practice there are several accompanying problems: insignificant sample overlap, lack of unique identifiers, confidentiality and expense.

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



INTERRECORD DISSIMILARITY MEASURE (DISTANCE FUNCTION):

$$C_{IJ} = F(P1_I, \ldots, P_{RI}, Q1_J, \ldots, Q_{RJ})$$

Fig. 1. Statistical file merging.

in both sample files. Thus, even if exact matching were not in violation of the confidentiality strictures, the information gain for policy purposes would be insignificant.

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. 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.

But even in situations where exact matching is possible, it is often precluded by confidentiality and cost considerations. In many instances privacy legislation guarantees respondents that information given for one file will not be used to "check up" on information given for another file. It may also be significantly more costly to achieve an exact match than a statistical match since, 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.

4.2. Statistical and constrained merges

Matching data files with the restriction that the means and variance–covariance matrix of data items in each file be fully retained in the matched file is designated as *constrained matching*. The equivalence of this restriction to the addition of a series of constraints to the merge process will be developed in subsequent sections. Examples of constrained matching are given by Budd [10] and by Turner and Gilliam [23]; see [22] for a history and survey of statistical matching.

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 purposes 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:

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

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

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

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

4.3. The assignment model of a constrained merge

Each microdata record consisting of r items can be viewed as a point in a 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:

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

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

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

where

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

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

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

$q_{jk} \equiv$ value of the k^{th} 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 [4]) to find a set of x_{ij} values that minimize expression (4) while satisfying constraints (5), (6) and (7).

In this model, originally posed in [24], c_{ij} is a measure of inter-record dissimilarity based on a comparison of corresponding record attributes. The specification of this function is dependent upon the statistical properties of the data items p_{ik} and q_{jk} and, given certain distributional assumptions, is uniquely determined (see [16]). Thus, the parameter c_{ij} can be viewed as the "distance" between record i of file A and record j of file B, and the problem of determining

a set of x_{ij} values that minimize the aggregate distance between matched records also yields the assignment problem.

Consider a pair of files with two common attributes: wages and salaries earned (p_{i1} and q_{j1}), the sex of the reporting unit (p_{i2} and q_{j2}). A simplistic distance function might take the form:

$$c_{ij} = w_1|p_{i1} - q_{j1}| + w_2 s_{ij},$$

where $s_{ij} = 0$ if $p_{i2} = q_{j2}$, else $s_{ij} = 1$; and w_1 and w_2 are weights reflecting the relative importance and magnitude of the respective items. While a unique measure has been derived, in practice distance functions are designed to emphasize those items of importance to the merge file user. In either case, a file obtained using the assignment model or the following transportation formulation is said to be an *optimal constrained match*, as it has been optimized with respect to a given distance function.

4.4. 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 i^{th} 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:

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

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

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

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 in [10, 22].

If c_{ij} is specified as 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 (8), then the problem becomes:

$$\text{minimize} \quad z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}, \tag{12}$$

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

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

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

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 [11]. This last observation is extremely important for computational reasons, as described in a subsequent section.

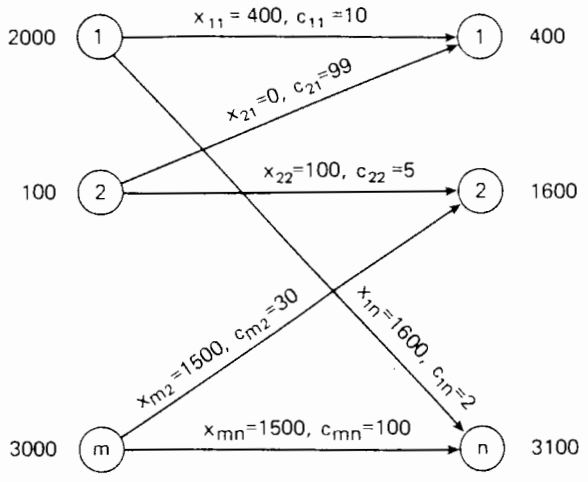
The dual problem for this model is:

$$\text{maximize} \quad w = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j, \tag{16}$$

$$\text{subject to} \quad u_i + v_j \leq c_{ij} \quad \text{for all } i \text{ and } j, \tag{17}$$

$$u_i, v_j \text{ unrestricted in sign.} \tag{18}$$

The analogy between this formulation of the merge process and the transportation network model described earlier in this volume also provides an intuitively appealing means of visualizing the underlying common problem. In the merge model analogy depicted to Fig. 2, the nodes represent individual microdata records whose weights are given as the supply and demand values.



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

Fig. 2. Example constrained merge as represented by transportation network model

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 objective is to determine the set of record matches and associated weights such that the original record weight totals are maintained at a minimum overall distance.

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. (See appendix for proofs). Therefore, this optimal merging not only minimizes aggregate information loss in the matching process, but preserves the complete statistical structure of the original files, two important characteristics missing from all other available merging schemes.

Unconstrained matching of two microdata files is given by applying either constraints (13) or (14) but not both. In this case the item means and variance–covariance matrix of only one of the files is preserved in the matching process. Okner [21] describes an example of unconstrained matching which is the model of (12), (13), and (15). See [7] for a critique of unconstrained matching.

The transportation model for optimal constrained microdata matching was originally posed in [24] and further discussion is given in [23]. A theoretical formulation of an optimal constrained merging is given in Kadane [16], where it is corroborated that under certain conditions constrained matching is analytically equivalent to the transportation model.

5. 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.

Since 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, a large-scale network solution system for the optimally-constrained merge problem was developed. This Extended Transportation System (ETS) makes use of recent research into network solution techniques [2, 5, 6, 13–15, 18, 20] and is based on a specialization of the primal simplex method. This system has been

used to solve some of the largest known optimization problems and is the only file merge system of its kind in existence.

5.1. Computational aspects of the primal simplex method

The primal simplex method as specialized to transportation problems has many computational advantages over other approaches. First, a simplex basis for an $m \times n$ problem corresponds to a spanning tree with $(m + n - 1)$ arcs. As such, a basis can be represented compactly using lists of node labels and corresponding flow values. This same data structure carries the basis inverse implicitly and, in conjunction with a set of node potentials and list structures for their maintenance, dramatically streamlines the simplex pricing and pivoting steps. It is through these elegant mathematical structures that the superior efficiencies, in terms of solution speed and memory requirements, are attained by this approach.

It is also important to note that, in contrast to out-of-kilter and primal–dual approaches which require all problem data to be in primary storage, only the basic arc data need be so maintained. In addition, the arc cost/distance data may be inspected piecemeal and can therefore reside on a secondary storage device and inspected in pages, or blocks of data. Identification of efficient rules for paging and pivot selection has been the subject of much research [9, 13, 14, 18, 20].

Another valuable characteristic of network problems in general is the automatic integrality of variables when all supplies and demands are whole numbers. When the distance data are also integer-valued, no program data need be represented as real numbers with the attendant concerns of numerical round-off and error tolerances.

5.2. The ETS solution system

In designing a network solution system for the OTA merge problem, the hardware available was a UNIVAC 1108 with only 150 000 36-bit words of 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 paged-data primal-simplex network code [18] because of the need to maintain in primary storage a basis of size $(6m + 6n)$ words plus a page of arc distance data. Even when the problem specifications were reduced to 50 000 constraints and 65 million variables, primary storage was insufficient.

The result was a twofold problem: first, the major data processing task of efficiently handling the arc distance data and, secondly, the extension of network solution technology to a new level to accommodate problems of this magnitude. The following sections describe ETS features designed to meet these needs.

5.2.1. Transportation problem optimizer

The primal simplex transportation code with the smallest known memory requirements is used. ETS employs a modification of the SUPERT code by Barr [2] which stores the 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 storing the program and pages of the arc distance data. It should be noted, however, that this condensed storage technique markedly increases the computational burden associated with the execution steps since every reference to problem data requires either a packing or an unpacking operation. Preliminary testing indicated that, as a result, solution times have been increased by a factor of from two to four over normal, unpacked data storage.

Partially offsetting this implementational disadvantage is the high efficiency of the transportation optimizer. The SUPERT code uses an independently derived variant of the ATI algorithm [14] and compares favorably with state-of-the-art primal simplex network codes. As shown in Table 1, in a comparison of aggregate solution times on a standard set of small transportation problems [19], the PNET-I code [15] is 73% slower than SUPERT, the GNET code [9] requires 106% more time, and the ARC-II code [5] is roughly comparable. Besides these primal simplex-based codes, the times for the SUPERK out-of-kilter code [3] are over five times those of SUPERT. These network codes have the disadvantage of being designed for more general capacitated problems but have the advantage

Table 1
Total solution times on transportation problems on a CDC 6600 [a]

NETGEN Problem	m	n	Arcs	SUPERT[b]	PNET-I[c]	GNET[d]	ARC-II[c]	SUPERK
1	100	100	1300	0.42	0.92	1.06	0.60	3.72
2	100	100	1500	0.59	0.98	1.08	0.68	4.25
3	100	100	2000	0.70	1.20	1.45	0.76	4.39
4	100	100	2200	0.70	1.07	1.44	0.68	4.27
5	100	100	2900	0.85	1.61	1.76	0.90	4.23
6	150	150	3150	1.29	2.28	2.45	1.60	7.09
7	150	150	4500	1.70	2.79	3.39	1.62	8.11
8	150	150	5155	1.95	3.11	4.06	2.17	8.61
9	150	150	6075	2.05	3.29	4.12	2.11	DNR
10	150	150	6300	2.04	4.08	4.68	2.81	DNR
Total time:				12.29	21.33	25.49	13.93	

DNR = Did not run in $201\,000_8$ words of memory.

[a] All programs compiled under FTN with OPT = 2. Times are elapsed CPU time exclusive of input and output.

[b] Modified row most negative pivot strategy used (see [14]).

[c] Modified node most negative pivot strategy used (see [13]).

[d] Default pivot strategy used.

of more advanced data structures. These programs also have substantially greater memory requirements than the ETS transportation optimizer.

5.2.2. Nondense problem generation

The *density*, d, of a transportation problem is defined as the number of problem arcs divided by (mn), the number of arcs if all origin–destination pairs are considered. Because of the enormous size of (mn) in the merge model, problems with $d < 1$ 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.

Specifically, one scheme narrows the window of consideration to the t records in file B that match most closely a given record in file A, based on one or more common attributes. This has been used with $t = 500$ and 1000, with the attribute being adjusted gross income. Since the merge file in this case was used in tax policy models, the income attribute was deemed to be of key importance; however, the size of the window still allows the various other factors, as expressed in the distance function, to influence the match process.

5.2.3. Distance function scaling

The range of the distance function values is reduced to 64 categories to permit exploitation of the machine wordsize by the data packing scheme described above. This is necessitated by a worst-case analysis of 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. But even with this scaling, a sufficient degree of distance value differentiation is available to produce an excellent match for the problems under consideration (see Section 5.4 regarding match quality).

5.2.4. Phase 1/phase 2 solution strategy

Initially the construction of a feasible basis is attempted from a single pass of the problem variables. If a feasible solution is not found, artificial arcs are added to form the starting basis and must be purged by the solution process. The wordsize restriction 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 $(d = 1)$, these artificial variables correspond to legitimate matching possibilities that fell outside of one record's window. 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.

This approach is a costly one, time-wise, as demonstrated in Section 5.3; however, OTA deemed merged file quality to be more valuable than the

additional machine time. The effect of allowing variables to remain in the merged file have not been investigated.

5.2.5. Closeness to optimality calculations

Two new procedures are incorporated in ETS to compute "closeness to 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 structure of transportation problems can be exploited to expedite calculation of such a lower bound. Both algorithms are detailed in [1, 8] and the more successful, in terms of strength of the bound, will now be presented.

From duality theory it is known that, for any feasible solution $\{x_i\}$ to the primal problem (12)–(15) with value z and any feasible solution $\{u_i, v_j\}$ for the dual problem (16)–(18) with value w, the relationship $w \leq z$ holds. Moreover, $w \leq w^* = z^* \leq z$, where w^* and z^* are the optimal solution values for the dual and primal problems. Hence, the objective function value w for any dual feasible solution is a lower bound on the optimal solution value. The following algorithm constructs just such a solution and bound from a primal feasible transportation basis.

For each primal feasible solution to the transportation problem, the simplex method associates a dual solution $\{u_i, v_j\}$, the node potentials. If the primal solution is not optimal, then the dual solution is not feasible and one or more (nonbasic) arcs violate constraint (17). In particular, if arc (i, j) is dual infeasible,

$$\Pi_{ij} = u_i + v_j - c_{ij} > 0.$$

If node potential u_i is decreased in value by Π_{ij}, the arc (i, j) becomes dual feasible. This new dual solution is obtained by a change of variables using the relation $u_i' = u_i - \Pi_{ij}$, which yields

$$u_i' + v_j = c_{ij}.$$

Moreover, the dual feasibility of any other arc (i, k) out of node i is not altered. Since $\Pi_{ij} > 0$, if $u_i + v_k < c_{ik}$, then

$$u_i' + v_k = (u_i - \Pi_{ij}) + v_k < c_{ik}.$$

No other arcs are affected by this change of dual variables.

The result of this substitution is a dual solution with at least one fewer dual infeasible arc with the new objective function value $w' = w - a_i\Pi_{ij}$. This pro-

cedure 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 z^*.

While this bound requires substantial processing to calculate, the bound becomes quite strong as intermediate solutions approach the optimal. This was verified by testing on medium-sized (250 000 arc) problems. Because of the speed of ETS, however, all production problems have been run to optimality. In other instances where greater machine time restrictions exist, this bound can be used to evaluate the quality of a suboptimal solution.

5.2.6. Pricing strategies

The pricing procedure is enhanced through the use of a 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 [13, 20, 25]. This tactic scans a page of arc data for pivot-eligible arcs ($\Pi_{ij} > 0$) and generates a "candidate list" of such arcs with predefined length l. The arc with the largest Π_{ij} value is selected, removed from the list, and pivoted into the basis. The remaining arcs in the list are then repriced. The "most eligible" candidate arc is selected from the revised list and the process continues until k such candidates are chosen or all candidate arcs on the list price nonpositive. At that point the list is replenished and the process repeated. This continues until the entire page of arc data prices dual feasible or until s passes of the page have been made. When all pages price nonpositive, optimality has been achieved. The selection of values for the parameters k, l, and s determine the effectiveness of the pricing procedure.

It should also be noted that all arc data input is "double-buffered", a systems programming technique which permits the pricing and pivoting operations to be carried out simultaneously with the paging in of arc distance data. In this manner, the central processing unit will rarely have to wait for a subsequent page of data to be read into primary storage from disk.

5.2.7. Other ETS *implementation aspects*

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 [17].

ETS also includes the capability for resuming the optimization process from a suboptimal solution, a command language for execution control, and report generation options.

5.3. *Recent* ETS *usage*

In order to assess the impact of tax rebate proposals and President Carter's tax reform initiatives, a merge of the 1975 SOI file and Survey of Income and Education (SIE) file (a one-time survey, equivalent to the CPS) was performed in the fall of 1978. The results were used in the preparation of [27]. Similar files have been used in the past to analyze former Secretary William Simon's fundamental tax reform proposals, the results of which appeared in [26].

Because of the enormity of the problem (110 094 constraints), the merge was broken into six subproblems based on census region. Each subproblem was optimized and the ETS solution statistics for these runs are given in Table 2. 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 [11]. Using this figure as a basis of comparison, a general-purpose mathematical programming system running on a dedicated UNIVAC 1108 would require over seven months to solve these problems.

The values in Table 2 show that phase 1 required approximately one-third of the solution time to drive out artificial variables constituting an average of 6.4% of the initial basis. This is also an indication of the time that could potentially be saved by the Big M method or by the construction of an initial primal feasible solution.

The "percent degenerate pivots" figures show that these transportation problems have relatively little degeneracy, a characteristic noted in studies of smaller transportation problems. This is in sharp contrast with assignment and transshipment network problems which have been shown to exhibit over 95 and 80 percent degenerate pivots, respectively [4, 6, 13].

A more curious finding from these statistics is that the number of pivots is highly correlated with the number of constraints ($\rho^2 = 0.92$) but not with the number of variables ($\rho^2 = 0.06$). This may indicate that a much larger window could be used in the problem generator without drastically escalating the solution times.

5.4. *Quality of the merged file*

Properly assessing the quality of a merge file is a difficult task since no generally accepted measures of "goodness" have been established and the theory in this area has only recently begun to be investigated. (The derivation of measures of match quality and their interrelationships with distance function

Table 2
ETS run statistics for six example subproblems

	Problem number							
	1	2	3	4	5	6	Total	Average
No. of constraints	15 002	17 897	24 946	13 946	15 507	22 796	110 094	18 349
No. of SOI records	9 406	9 664	9 447	7 361	7 487	6 635	50 000	8 333
No. of SIE records	5 596	8 233	15 499	6 585	8 020	16 161	60 094	10 016
No. of variables	4 703 000	4 832 000	4 723 500	3 680 500	3 743 500	3 317 500	25 000 000	4 166 667
Solution time[a]								
in minutes	285	382	780	254	323	598	2622	437
in hours	(4.75)	(6.36)	(13.01)	(4.24)	(5.39)	(9.97)	(43.7)	(7.28)
Artificials at start	991	1 088	1 962	732	1 128	1 190	7 091	1 182
Time spent in phase 1	28%	27%	37%	31%	42%	38%	35%	34%
No. of pivots performed	197 202	236 931	391 289	162 439	217 164	288 834	1 493 859	248 997
Percent degenerate pivots	1.6%	1.7%	2.6%	1.5%	2.8%	2.4%	2.2%	2.1%

[a] Central processor time on UNIVAC 1108 with system written in and compiled under FORTRAN V level 11A.

definitions are important topics for future research.) Somewhat simplistic measures can be used, however, to give a broad-brush indication of the degree of agreement between the records joined to form the composite file. To this end, Tables 3 and 4 provide summary information regarding the merge file described above.

As depicted earlier in Fig. 1, a composite record is formed by mating a record in file A with a record in file B and assigning a record weight. This record then contains duplicate items since some attributes appear in both original files. These common items are, of course, used in the distance function calculation but specific values can also be compared to see how well individual records matched.

Table 3 shows percentages of agreement and average differences between like items in the composite records. For example, 95.1 percent of the merged records had the same I.R.S. tax schedule code. These measurements are calculated using the record weights, so as to reflect the degree of agreement for the merged populations rather than the matched samples.

These figures indicate that by minimizing the aggregate distance function values and maintaining the record weight constraints, that a very strong match can be obtained. It should be noted that 100 percent agreement between items is virtually impossible since the match is between different samples. For example,

Table 3
Item analysis of the complete merged file

Common date item	Matched records relationship	Weighted percentage[a] of records or value
1. Schedule code (single, joint, married filing separately, etc.)	% agreement	95.1%
2. Age of tax filer	% within 5 years	60.9%
	% within 10 years	91.7%
3. Size of family	% agreement	70.2%
	% within 2	97.4%
4. Race	% agreement	89.3%
5. Sex	% agreement	94.6%
6. Adjusted gross income (including all taxable sources of income)	average difference	$925
	% within $1000	79.6%
	% within $2000	92.6%
7. Wages and salaries	average difference	$637
	% within $1000	86.7%
	% within $2000	95.0%

[a] Percentages based on sums of record weights with indicated agreement as a percentage of the total of all record weights.

Table 4
Composite agreement count for six common items in the complete merged file

Number of item agreements[a]	Percent of records (weighted)[b]	Cumulative percent (weighted)[b]
6	68.6%	68.6%
5	22.0	90.6
4	6.4	97.0
3	2.1	99.1
2	0.6	99.8
1	0.2	100.0
0	0.0	100.0

[a] Categories of item agreement in a composite record: (1) same schedule codes; (2) ages within ten years; (3) family size within two; (4) same race; (5) same sex; and (6) adjusted gross income within $2000.

[b] Percentage based on sums of record weights exhibiting such agreement as a percentage of the total of the record weights, 82 215 537 (the number of tax filers).

if the match were made on the basis of schedule code alone and all constraints relaxed, the best possible level of agreement would be 98.2 percent.

To identify record agreement on multiple items, six agreement categories were defined, the number of categories of agreement for each record counted, and the results summarized in Table 4. Again using weighted counts, 68.6 percent of the merge file records agree in all six categories and over 90 percent agree in five or more categories. Therefore, this particular file not only is a good match on individual items but on combinations of items as well.

Postmerge calculations also verified the retention of the statistical structure of both original files' data. Note that while the figures in Tables 3 and 4 could be improved by relaxing either constraints (13) or (14), this would yield distortions in the aggregate statistics for all data items from the corresponding original file. Such distortions could significantly alter the results obtained by the personal income tax and transfer income models.

6. Summary

Whereas separate surveys for different informational needs would cost tens of millions of dollars apiece, this optimal, constrained 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 to be a cost-effective means of providing new, high-quality data resources for the public decision-making process.

Appendix. Preservation of item statistics in constrained merging

In this section we show that the means and variance–covariance matrix 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 and the inclusion of all of the original items from both files in the composite file. (See Fig. 1.) This discussion does not apply to any relationships between items that were originally in different files.

A.1. 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 r^{th} data item in the i^{th} 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

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

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

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

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

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

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

A.2. Variance–covariance matrices

For a similar analysis of the items' variance–covariance properties, let p_{ir} and

p_{is} represent, respectively, the r^{th} and s^{th} data items in the i^{th} record of file A. The following expression defines σ_{rs}^2 as the variance of item r (if $r = s$) or the covariance of the two items (if $r \neq s$) in the original file:

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

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

$$\sigma_{rs}^{2*} = \left\{\sum_{i=1}^{m}\sum_{j=1}^{n} [x_{ij}(p_{ir} - p_r^*)(p_{is} - p_s^*)]\right\} \Big/ \left(\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}\right).$$

Since $p_r^* = \bar{p}_r$ and $p_s^* = \bar{p}_s$.

$$\sigma_{rs}^{2*} = \left\{\sum_{i=1}^{m} [(p_{ir} - \bar{p}_r)(p_{is} - \bar{p}_s)\left(\sum_{j=1}^{n} x_{ij}\right)\right\} \Big/ \left(\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\right)$$

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

which is equivalent to σ_{rs}^2. 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).

Acknowledgment

Thanks are given to Darwin Klingman, University of Texas at Austin and David Karney, Williams Companies, for their valuable suggestions and contributions to this paper. We also wish to acknowledge Alan J. Goldman, John M. Mulvey and the referees, whose critiques led to a much-improved paper. Finally, we thank Harvey Galper, Nelson McClung and Gary A. Robbins of the Office of Tax Analysis for their strong interest in and support of this project.

References

[1] Analysis, Research and Computation, Inc., "Extended Transportation System (ETS) programmer technical reference manual", P.O. Box 4067, Austin, TX (1975).

[2] Richard S. Barr, "Primal simplex network codes: A computational study", Research Report, Edwin L. Cox School of Business, Southern Methodist University, Dallas, TX (1980).

[3] Richard S. Barr, 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) 60–86.

[4] Richard S. Barr, Fred Glover and Darwin Klingman, "The alternating basis algorithm for assignment problems", *Mathematical Programming* 13 (1977) 1–13.

[5] Richard S. Barr, Fred Glover and Darwin Klingman, "Enhancements to spanning tree labelling procedures for network optimization", *INFOR* 17 (1) (1979) 16–33.
[6] Richard S. Barr, Joyce Elam, Fred Glover and Darwin Klingman, "A network augmenting path basis algorithm for transshipment problems", in: A.V. Fiacco and K.O. Kortanek, eds., *External methods and systems analysis* (Springer, Berlin, 1980).
[7] Richard S. Barr and J. Scott Turner, "New techniques for statistical merging of microdata files", in: R. Haveman and K. Hollenbeck, eds., *Microeconomic simulation models for public policy analysis* (Academic Press, New York, 1980).
[8] Richard S. Barr and J. Scott Turner, "A new, linear programming approach to microdata file merging", in: U.S. Department of the Treasury, *1978 Compendium of tax research* (U.S. Government Printing Office, Washington, D.C., 1978) pp. 129–155.
[9] Gordon H. Bradley, Gerald G. Brown and Glenn W. Graves, "Design and implementation of large scale primal transshipment algorithms", *Management Science* 24 (1) (1977) 1–34.
[10] Edward C. Budd, "The creation of a microdata file for estimating the size distribution of income", *Review of Income and Wealth* 17 (4) (1971) 317–334.
[11] A. Charnes and W.W. Cooper, *Management models and industrial applications of linear programming* (Wiley, New York, 1961).
[12] Fred Glover, John Hultz and Darwin Klingman, "Improved computer-based planning techniques, Part 1", *Interfaces* 8 (4) (1978) 16–25.
[13] Fred Glover, David Karney and Darwin Klingman, "Implementation and computational comparisons of primal, dual and primal–dual computer codes for minimum cost network flow problems", *Networks* 4 (3) (1974) 192–211.
[14] Fred Glover, 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) 793–813.
[15] Fred Glover, Darwin Klingman and Joel Stutz, "Augmented threaded index method for network optimization", *INFOR* 12 (3) (1974) 293–298.
[16] Joseph Kadane, "Some statistical properties in merging data files", in: U.S. Department of the Treasury, *1978 Compendium of tax research* (U.S. Government Printing Office, Washington, D.C., 1978).
[17] James A. Kalan, private communication.
[18] David Karney and Darwin Klingman, "Implementation and computational study on an in-core/out-of-core primal network code", *Operations Research* 24 (6) (1976) 1056–1077.
[19] D. Klingman, A. Napier and J. Stutz, "NETGEN' A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems", *Management Science* 20 (5) (1974) 814–821.
[20] John M. Mulvey, "Pivot strategies for primal-simplex network codes", *Journal of the Association for Computing Machines* 25 (2) (1978) 266–270.
[21] Benjamin Okner, "Constructing a new data base from existing microdata sets: The 1966 merge file", *Annals of Economic and Social Measurement* 1 (1972) 325–342.
[22] Daniel B. Radner, "The development of statistical matching in economics", *1978 Proceedings of the American Statistical Association, Social Statistics Section* (1978).
[23] J. Scott Turner and Gary B. Gilliam, "Reducing and merging microdata files", OTA Paper 7, Office of Tax Analysis, U.S. Department of the Treasury, Washington, D.C. (1975).
[24] J. Scott Turner and Gary A. Robbins, "Microdata set merging using microdata files", Research Report, Office of Tax Analysis, U.S. Department of the Treasury, Washington, D.C. (1974).
[25] V. Srinivasan and G.L. Thompson, "Benefit-cost analysis of coding techniques for the primal transportation algorithm", *Journal of the Association for Computing Machinery* 20 (1973) 194–213.
[26] U.S. Department of the Treasury, *Blueprints for basic tax reform* (U.S. Government Printing Office, Washington, D.C., 1978).
[27] U.S. Department of the Treasury, *The President's* 1978 *tax program* (U.S. Government Printing Office, Washington, D.C., 1978).