PSYCHOMETRIKA-VOL. 46, NO. 4
DECEMBER, 1981

MATRIX REORGANIZATION AND DYNAMIC PROGRAMMING: APPLICATIONS TO PAIRED COMPARISONS AND UNIDIMENSIONAL SERIATION

L. J. HUBERT AND R. G. GOLLEDGE

THE UNIVERSITY OF CALIFORNIA, SANTA BARBARA

A recursive dynamic programming strategy is discussed for optimally reorganizing the rows and simultaneously the columns of an $n \times n$ proximity matrix when the objective function measuring the adequacy of a reorganization has a fairly simple additive structure. A number of possible objective functions are mentioned along with several numerical examples using Thurstone's paired comparison data on the relative seriousness of crime. Finally, the optimization tasks we propose to attack with dynamic programming are placed in a broader theoretical context of what is typically referred to as the quadratic assignment problem and its extension to cubic assignment.

Key words: combinatorial optimization, quadratic assignment, cubic assignment.

One of the major difficulties shared by almost all data analysis procedures with some type of combinatorial optimization component is the enormous computational burden that is imposed when optimal solutions are sought (e.g., see the methods discussed by Arabie & Carroll, 1980; Defays, 1978; Holman, 1978; Flueck & Korsh, 1974). In fact, a very active area of computer science has grown up around the contention that many of these optimization problems are inherently so hard that really good algorithms will probably never be found [Karp, 1972]. This is true even for some of the well-known tasks of reorganizing dichotomous (zero-one) paired-comparison matrices of the type discussed extensively in the psychometric literature [Baker & Hubert, 1977].

Given the rather negative implications of this newer work in computer science toward discovering best possible combinatorial optimization algorithms for specific problems, it is not surprising that heuristic procedures leading to good but not necessarily optimal solutions have become very important for handling moderate to large problems. Still, without methods that can check on the performance of these heuristics, much of the resulting literature must rest on speculation and intuitive justification. Procedures that can provide exact solutions for relatively small problems are important, if only to generate optimal solutions that can then be used as baselines to evaluate the adequacy of a given heuristic.

We emphasize one representative combinatorial optimization task, defined by the optimal reorganization of the rows and simultaneously the columns of an $n \times n$ matrix **A**. The natural psychometric reference is to paired comparison matrices but the approach we take has significant data analysis implications for the more general problem of unidimensional seriation or scaling.

For notational purposes, let an arbitrary entry from the i^{th} row and j^{th} column of **A** be denoted by a_{ij}; it is assumed that the main diagonal is irrelevant and consists of all zeros, i.e., $a_{ii} = 0$, $1 \leq i \leq n$. Thus, if $\rho(\cdot)$ is some permutation of the first n integers, i.e., ρ: (1, 2, ..., n)$\rightarrow$ (1, 2, ..., n), then a particular reorganization of **A** based on relabeling the rows and columns by the function $\rho(\cdot)$ can be represented as $\{a_{\rho(i)\rho(j)}\}$. Our task is, first, to assign some measure of adequacy (i.e., the value for some objective function) to each of the $n!$

Partial support for this research was provided by NIJ Grant 80-IJ-CX-0061.

Requests for reprints should be sent to either L. J. Hubert, Graduate School of Education, University of California, Santa Barbara, California 93106 or R. G. Golledge, Department of Geography, University of California, Santa Barbara, California 93106.

0033-3123/81/1200-3064 $00.75/0

reorganizations of **A** and then locate those permutations that maximize (or possibly minimize) this index. In the paired comparison context, for example, a_{ij} could denote the proportion of times i was chosen over j; one natural measure of adequacy in this context is the sum of the entries above the main diagonal.

Depending on the particular objective function used, it may be possible to develop a recursive dynamic programming approach to the problem of optimal matrix reorganization. The computational burden is still extensive, but is much less than what would be required by an evaluation of the chosen objective function over all $n!$ possible permutations. We emphasize one simple dynamic programming strategy that can be used for a variety of objective functions that have a rather elementary form and introduce it in the paired comparison framework. Once this general strategy is laid out we present a number of possible applications and an example.

Background: Paired Comparison Methods

The most direct method for locating a best reorganization of a matrix **A** is by complete enumeration. The $n!$ realizations of **A** are generated by relabeling the rows and columns in all possible ways; those realizations that attain the best value for the objective function identify global optima. In brief, a dynamic programming approach for the same problem has value as a parsimonious solution strategy for it reduces the enumeration from a consideration of all permutations ($n!$) to a consideration of 2^n possible subsets. (As a rough comparison of the discrepancy between $n!$ and 2^n using Stirling's approximation, $n!/2^n \approx \sqrt{2\pi n}\,(n/2e)^n$. Obviously, this latter term increases at a very fast rate. Even when n is as small as 9, $n!$ is about 700 times as large as 2^n.) The number of subsets can still be very large and the biggest matrices that can be dealt with conveniently, using the proposed dynamic programming strategy, are about 15×15. Beyond this size, the storage and computational requirements become excessive very quickly. Nevertheless, the alternative of evaluating an objective function over all 15! realizations of **A** is clearly out of the question; in fact, it is doubtful whether complete enumeration beyond an n of 9 should ever be considered a reasonable alternative.

To give a very simple example based on a strategy originally due to Lawler [1964] that can be generalized very nicely, suppose **A** is asymmetric and the objective function or measure of adequacy we wish to maximize is the sum of entries above the main diagonal. The typical application would be to a paired comparison matrix in which a_{ij} represents the proportion of subjects who rate object i "better than" object j; the reorganization of **A** merely locates the n objects along a single dimension. The constraint that the corresponding off-diagonal entries sum to 1, i.e., $a_{ij} + a_{ji} = 1$, is convenient in our initial discussion of paired comparison matrices and will be assumed. The optimization procedure, however, can also be applied to matrices that do not satisfy this condition if the particular application so warrants.

The illustration just given using the sum of the above diagonal entries of **A** as an objective function forms the basis of a substantial literature in psychology and elsewhere. Depending on how the entries in **A** are defined, this optimization task has been referred to, among other names, as maximum likelihood paired comparison ranking [Flueck & Korsh, 1974], triangulating an input-output matrix [Korte & Oberhofer, 1971], and finding minimum feedback arc sets [Lawler, 1964]. For a review, see Hubert [1976].

As some further notation, suppose S denotes an n-element object set $\{0_1, 0_2, \ldots, 0_n\}$ and R some arbitrary subset of S, with the interpretation that the m elements in R in some order form the first m rows and columns of a reorganized matrix. The best value of the objective function when only the elements in R are considered is denoted by $f(R)$, i.e., the optimal value for the sum of above diagonal entries using the $m \times m$ submatrix of **A**

constructed from the m elements in R. Thus, if we wish to evaluate $f(R')$ for some subset with $m + 1$ members, the simple recursion formula given in (1) can be used:

$$f(R') = \max[f(R) + d(R, 0_k)],$$

where the maximum is taken over all subsets R and objects 0_k such that $R \cup \{0_k\} = R'$, and $d(R, 0_k)$ is the sum of proximities from the objects in R to 0_k. Based on this recursion and starting from the initial condition that $f(\varnothing) = 0$, where $\varnothing$ is the empty set, $f(S)$ is the optimal value for the objective function, i.e., the maximum sum of above diagonal elements. An actual reordering that provides this best value can be found by working "backwards" from $f(S)$, i.e., first finding the last object placed, then the second to the last, and so on. A small example of this process is developed later in the paper.

The problem of maximizing the sum of above diagonal entries can be solved through dynamic programming because of a very nice property for the objective function being used: given that the m objects in R are placed in the first m rows and columns, the contribution to the final value for the objective function of another object, 0_k, placed at the position of the $m + 1^{st}$ row and column [i.e., what is called $d(R, 0_k)$ in (1)] is additive and does not depend on how the objects in R are ordered. Thus, *any* objective function that we choose with a similar additive structure and invariance property with respect to R can also be solved by dynamic programming through a recursion of the form given in (1). Since the function $f(\cdot)$ is defined implicitly by the recursion, however, and it constructed in stages, the values of $f(\cdot)$ at any given stage may not be as simple to interpret as in this first example. In any event, the recursion in (1) demands an evaluation of $f(\cdot)$ over all 2^n possible subsets of S; thus, when n is even moderate in size, this number is substantially less than that required by a complete enumeration of all possible orderings of **A** (Unless otherwise stated, it is assumed that the objective function is to be maximized starting from the initial condition that $f(\varnothing) = 0$.)

A Simple Example

As a very small example of how the expression in (1) can be implemented for this simple objective function, consider the aggregate paired-comparison matrix given below on the seriousness of four crimes: Arson (0_1), Embezzlement (0_2), Kidnapping (0_3), and Seduction (0_4). These data were analyzed by Thurstone [1927] as part of a larger study.

	0_1	0_2	0_3	0_4
0_1	X	.348	.543	.716
0_2	.652	X	.752	.774
0_3	.457	.248	X	.086
0_4	.284	.226	.914	X

An entry in the i^{th} row and j^{th} column indicates the proportion of subjects who judged the column crime more serious than the row crime.

To locate that reordering of the rows and columns maximizing the above diagonal entries and by implication an ordering of the crimes from least to most serious, the recursion in (1) could be implemented as:

(i) $f(\varnothing) = 0$

(ii) $f(\{0_1\}) = 0; \quad f(\{0_3\}) = 0$

$f(\{0_2\}) = 0; \quad f(\{0_4\}) = 0$

(iii)
$$f(\{0_1, 0_2\}) = \max\{f(\{0_1\}) + d(\{0_1\}, 0_2), f(\{0_2\}) + d(\{0_2\}, 0_1)\}.$$
$$\max\{.348, .652\} = .652.$$
$$f(\{0_1, 0_3\}) = \max\{f(\{0_1\}) + d(\{0_1\}, 0_3), f(\{0_3\}) + d(\{0_3\}, 0_1)\}.$$
$$\max\{.543, .457\} = .543.$$
$$f(\{0_1, 0_4\}) = \max\{f(\{0_1\}) + d(\{0_1\}, 0_4), f(\{0_4\}\ 0_1)\}.$$
$$\max\{.716, .284\} = .716.$$
$$f(\{0_2, 0_3\}) = \max\{f(\{0_2\}) + d(\{0_2\}, 0_3), f(\{0_3\}) + d(\{0_3\}, 0_2)\}.$$
$$\max\{.752, .248\} = .752.$$
$$f(\{0_2, 0_4\}) = \max\{f(\{0_2\}) + d(\{0_2\}, 0_4), f(\{0_4\}) + d(\{0_4\}, 0_2)\}.$$
$$\max\{.774, .226\} = .774.$$
$$f(\{0_3, 0_4\}) = \max\{f(\{0_3\}) + d(\{0_3\}, 0_4), f(\{0_4\}) + d(\{0_4\}, 0_3)\}.$$
$$\max\{.086, .914\} = .914.$$

(iv)
$$f(\{0_1, 0_2, 0_3\}) = \max\{f(\{0_1, 0_2\}) + d(\{0_1, 0_2\}, 0_3), f(\{0_1, 0_3\}) + d(\{0_1, 0_3\}, 0_2), f(\{0_2, 0_3\}) + d(\{0_2, 0_3\}, 0_1)\}.$$
$$\max\{1.947, 1.139, 1.861\} = 1.947.$$
$$f(\{0_1, 0_2, 0_4\}) = \max\{f(\{0_1, 0_2\}) + d(\{0_1, 0_2\}, 0_4), f(\{0_1, 0_4\}) + d(\{0_1, 0_4\}, 0_2), f\{0_2, 0_4\}) + d(\{0_2, 0_4\}, 0_1)\}.$$
$$\max\{2.142, 1.290, 1.710\} = 2.142.$$
$$f(\{0_1, 0_3, 0_4\}) = \max\{f(\{0_1, 0_3\}) + d(\{0_1, 0_3\}, 0_4), f(\{0_1, 0_4\}) + d(\{0_1, 0_4\}, 0_3), f(\{0_3, 0_4\}) + d(\{0_3, 0_4\}, 0_1)\}.$$
$$\max\{1.345, 2.173, 1.655\} = 2.17.$$
$$f(\{0_2, 0_3, 0_4\}) = \max\{f(\{0_2, 0_3\}) + d(\{0_2, 0_3\}, 0_4), f(\{0_2, 0_4\}) + d(\{0_2, 0_4\}, 0_3), f(\{0_3, 0_4\}) + d(\{0_3, 0_4\}, 0_2)\}.$$
$$\max\{1.612, 2.440, 1.388\} = 2.440.$$

$$\text{(v)}\quad f(\{0_1, 0_2, 0_3, 0_4\}) = \max\{f(\{0_1, 0_2, 0_3\}) + d(\{0_1, 0_2, 0_3\}, 0_4),$$
$$f(\{0_1, 0_2, 0_4\}) + d(\{0_1, 0_2, 0_4\}, 0_3),$$
$$f(\{0_1, 0_3, 0_4\}) + d(\{0_1, 0_3, 0_4\}, 0_2),$$
$$f(\{0_2, 0_3, 0_4\}) + d(\{0_2, 0_3, 0_4\}, 0_1)\}.$$
$$\max\{3.523, 4.351, 2.995, 3.833\} = 4.351.$$

Thus, working backwards from $f(\{0_1, 0_2, 0_3, 0_4\})$, the best reordering is given by the matrix:

	0_2	0_1	0_4	0_3
0_2	X	.652	.774	.752
0_1	.348	X	.716	.543
0_4	.226	.284	X	.914
0_3	.248	.457	.086	X

indicating an ordering of the four crimes from least to most serious of Embezzlement, Arson, Seduction, and Kidnapping. We note that all of the entries above the main diagonal are greater than .500. Thus, for any pair of crimes, the obtained ordering is consistent in the sense that the least serious crime in the pair is placed to the left of the other.

A second way to represent the results of the recursion is given in Figure 1. Here, all $2^4 = 16$ subsets are used to form the nodes in a directed network. An edge between two nodes connects a pair of subsets that differ by a single object; the edge weights are the incremental contributions defined by $d(R, 0_k)$. The path traced by the node sequence has a sum of edge weights equal to 4.351 and defines the longest path in the network and the optimal reorganization identified above. In fact, all examples of the recursion in (1) could be reformulated as a longest path problem in such a network [cf. Dreyfus, 1969]. Direct use of the recursion in (1) merely bypasses this intermediate representation.

A Larger Example

As a more realistic illustration of what the matrix reorganization strategy does, Table 1 presents an aggregate paired comparison matrix which is also part of the data presented by Thurstone [1927] on the perceived seriousness of crime. The 15 crimes in this case are: Abortion (0_1), Adultery (0_2), Arson (0_3), Assault and Battery (0_4), Burglary (0_5), Counterfeiting (0_6), Embezzlement (0_7), Forgery (0_8), Homicide (0_9), Kidnapping (0_{10}), Larceny (0_{11}), Libel (0_{12}), Perjury (0_{13}) Rape (0_{14}), and Seduction (0_{15}). Table 2 presents an optimal reorganization based on maximizing the sum of above diagonal entries. Compared to the ordering of these 15 crimes using scale scores obtained by Thurstone ($0_{12}, 0_{11}, 0_4, 0_5, O_8, 0_6, 0_7, 0_{13}, 0_3, 0_2, 0_{10}, 0_1, 0_{15}, 0_9, 0_{14}$), there are some differences in detail. The optimal reordering of the matrix given in Table 2 reverses the pairs 0_7 and 0_{13}, 0_2, and $0_{10}, 0_1$ and $0_{15}, 0_9$ and 0_{14}. The sum of the above diagonal entries for the Thurstone order is 78.344, whereas in Table 2 it is 78.946.

The particular objective function used in defining what constitutes an optimal reordering is obviously relevant in determining how the objects should be placed along a continuum. Given reasonable well-structured data such as the Thurstone matrix of Table 1, however, the choice of objective function will typically affect only the fine structure of the

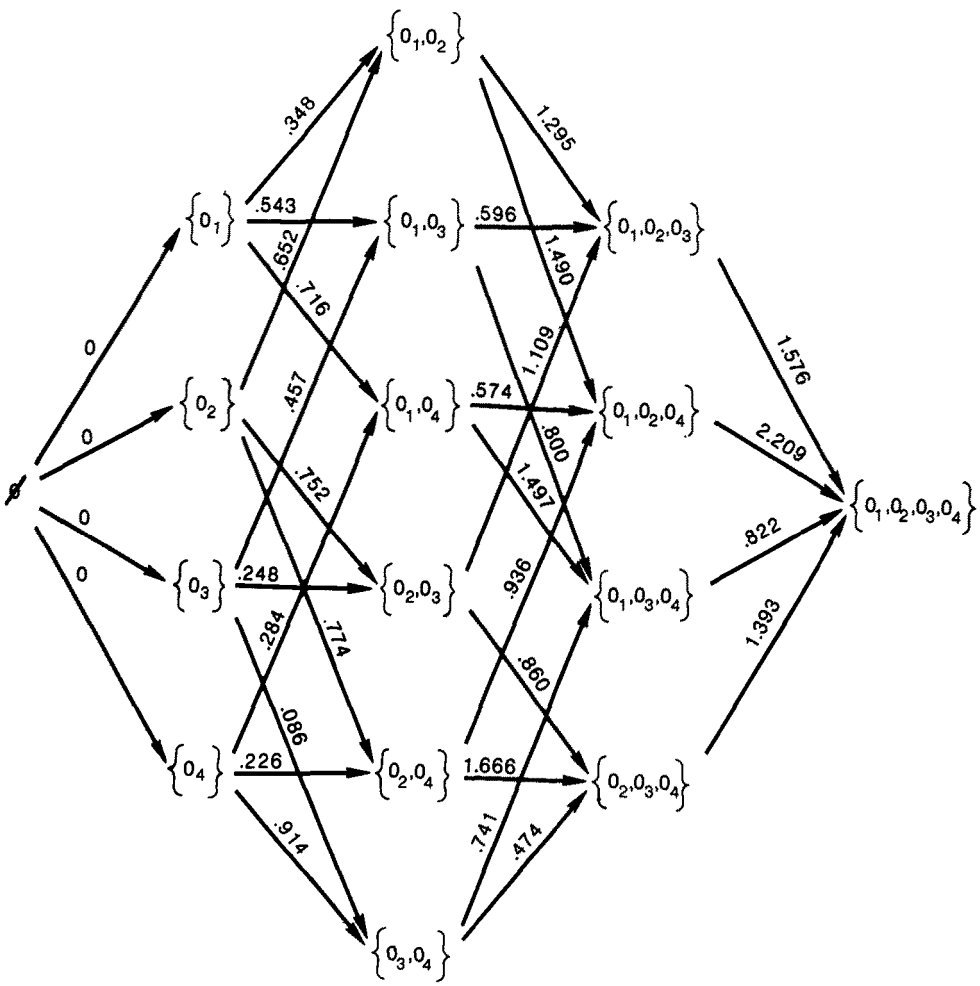

FIGURE 1.
Network Structure For the Simple 4 × 4 example Discussed in the Text

placement. We will see this again later when several other objective functions are considered. The obvious substantive implication is that any analysis of data subject to error may be more appropriately interpreted in terms of its general structure and not in terms of the detail offered by a particular model or procedure. All of the alternatives we discuss may be used to reorder some given matrix. However, none of these alternatives (including Thurstone's scaling strategy) could be considered the definitive standard.

A Comment on Computation

The network structure of Figure 1 suggests a convenient way of implementing the recursion in (1) and organizing the necessary computations efficiently for any definition of $d(R, 0_k)$ we choose to work with. Suppose two single dimension arrays are specified, each containing 2^n entries. There is one entry in each array for every subset and once a subset is known, the index it represents in binary form defines a unique place in either array. Assuming $f(R) = 0$ for $|R| = 1$ and our desire to maximize the chosen objective function, each subset of size r can be generated in turn (starting at $r = 1$) and used to update the values of $f(\cdot)$ for subsets of size $r + 1$. For instance, if R is the given size r subset, then $n - r$

TABLE 1

Paired Comparison Matrix from Thurstone [1927]

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}
O_1	X	.323	.338	.211	.238	.244	.245	.212	.760	.318	.222	.191	.256	.822	.419
O_2	.677	X	.415	.242	.281	.285	.253	.274	.863	.365	.207	.182	.245	.925	.589
O_3	.662	.585	X	.260	.226	.321	.348	.254	.917	.563	.215	.144	.349	.944	.716
O_4	.789	.757	.740	X	.515	.556	.485	.534	.970	.743	.385	.385	.587	.947	.785
O_5	.762	.719	.774	.485	X	.593	.605	.580	.981	.856	.333	.322	.478	.981	.769
O_6	.756	.715	.679	.444	.407	X	.540	.488	.947	.804	.303	.284	.532	.963	.756
O_7	.755	.747	.652	.515	.395	.460	X	.350	.958	.752	.305	.248	.474	.977	.774
O_8	.788	.726	.746	.466	.420	.512	.650	X	.951	.819	.343	.320	.534	.966	.820
O_9	.240	.137	.083	.030	.019	.053	.042	.049	X	.083	.030	.034	.079	.441	.181
O_{10}	.682	.635	.437	.257	.144	.196	.248	.181	.917	X	.170	.106	.288	.902	.595
O_{11}	.778	.793	.785	.615	.667	.697	.695	.657	.970	.830	X	.348	.648	.970	.848
O_{12}	.809	.818	.855	.615	.678	.716	.752	.680	.966	.894	.652	X	.702	.981	.886
O_{13}	.744	.755	.651	.413	.522	.467	.526	.466	.921	.712	.352	.298	X	.951	.767
O_{14}	.178	.075	.056	.053	.019	.037	.023	.034	.559	.098	.030	.019	.049	X	.076
O_{15}	.581	.411	.284	.215	.231	.244	.226	.180	.819	.405	.152	.114	.233	.924	X

supersets containing $r + 1$ members could be formed from it by adding a single object. If $f(R) + d(R, 0_k)$ is greater than the entry in the first array corresponding to the position for the subset $R \cup \{0_k\}$, then this former value replaces it in the array and the index k is placed in the corresponding position in the second array. Consequently, the first array is a means for continually updating the longest paths found thus far when a certain number of objects are already in position; the second array defines the objects last placed in the paths. Once all size r subsets have been used to update the array entries for the $r + 1$ subsets, we generate all subsets containing $r + 1$ members and use these to update the array information for the subsets defined by $r + 2$ elements, and so on.

The optimal value for the objective function is the array value in position 2^n after all subsets of size $n - 1$ have been generated and used to update the array entry for the single subset of size n. Furthermore, one path leading to this value can be retraced easily by using the entries in the second array starting from the 2^n position. We eliminate this object from the complete set S and locate it last in the sequence. The resulting subset of size $n - 1$ defines another object in the second array which is placed second to the last in the path. This object is then eliminated, defining a subset of size $n - 2$, and so on.

It should be noted that more than one path may exist that attains the optimal value. The procedures we have described, however, will not identify all of these unless the second array is expanded to include possible ties in the updating procedure. For objective functions such as the sum of above-diagonal entries and proximities that do not include a lot of ties, the optimal value will probably be attained for only one path. As we will see later, however, particular objective functions may be more prone to having optimal values that are multiply generated. In these instances, as well as more generally, care must be taken in

TABLE 2

Table 1 Reordered to Maximize the Sum of Above Diagonal Entries

	O_{12}	O_{11}	O_4	O_5	O_8	O_6	O_{13}	O_7	O_3	O_{10}	O_2	O_{15}	O_1	O_{14}	O_9
O_{12}	X	.652	.615	.678	.680	.716	.702	.752	.855	.894	.818	.886	.809	.981	.966
O_{11}	.348	X	.615	.677	.657	.697	.648	.695	.785	.830	.793	.848	.778	.970	.970
O_4	.385	.385	X	.515	.534	.556	.587	.485	.740	.743	.757	.785	.789	.947	.970
O_5	.322	.333	.485	X	.580	.593	.478	.605	.774	.856	.719	.769	.762	.981	.981
O_8	.320	.343	.466	.420	X	.512	.534	.650	.746	.819	.726	.820	.788	.966	.951
O_6	.284	.303	.444	.407	.488	X	.532	.540	.679	.804	.715	.756	.756	.963	.947
O_{13}	.298	.352	.413	.522	.466	.467	X	.526	.651	.712	.755	.767	.744	.951	.921
O_7	.248	.305	.515	.395	.350	.460	.474	X	.652	.752	.747	.774	.755	.977	.958
O_3	.144	.215	.260	.226	.254	.321	.349	.348	X	.563	.585	.716	.662	.944	.917
O_{10}	.106	.170	.257	.144	.181	.196	.288	.248	.437	X	.635	.595	.682	.902	.917
O_2	.182	.207	.242	.281	.274	.285	.245	.253	.415	.365	X	.589	.677	.925	.863
O_{15}	.114	.152	.215	.231	.180	.244	.233	.226	.284	.405	.411	X	.581	.924	.819
O_1	.191	.222	.211	.238	.212	.244	.256	.245	.338	.318	.323	.419	X	.822	.760
O_{14}	.019	.030	.053	.019	.034	.037	.049	.023	.056	.098	.075	.076	.178	X	.559
O_9	.034	.030	.030	.019	.049	.053	.079	.042	.083	.083	.137	.181	.240	.441	X

imparting too much importance to the fine detail of a given optimal ordering. Typically, optimal orderings, or these that are "close" to being optimal, will have a broad consistency even though minor variation will exist as to detail.

Other Objective Functions

A. Different Counting Rule

The sum of the above diagonal entries may be a natural measure to optimize when the matrix **A** is asymmetric, but there are many other possibilities for an objective function that could be considered. Given the general goal of forcing the large entries above the main diagonal and the small entries below, any measure of this differential is a candidate. For example, instead of the previous definition of $d(R, 0_k)$ as the sum of proximities from the objects in R to 0_k, $d(R, 0_k)$ could be specified as the number of times a proximity from R to 0_k is greater than a proximity from $\bar{R} - \{0_k\}$ to 0_k, where $\bar{R}$ is the complement of R. This last measure only depends on the rank order of the entries within a column and counts the number of times an above diagonal entry is greater than one below the main diagonal and within the same column. As before, the actual ordering of the elements in R or $\bar{R} - \{0_k\}$ is irrelevant and a dynamic programming strategy can be implemented as in (1) with $d(R, 0_k)$ defined appropriately. The function $f(\cdot)$ itself is defined recursively by the equation and the optimal value for the objective function is given by $f(S)$. As will be shown later, the use of this particular objective function defines an example of cubic assignment.

Going back to the data of Table 1, one possible optimal reordering of the Thurstone matrix based on this nonmetric criterion would be $(0_{12}, 0_{11}, 0_5, 0_6, 0_8, 0_7, 0_4, 0_3, 0_{13}, 0_{15},$

$0_{10}, 0_2, 0_1, 0_{14}, 0_9$). Obviously the choice of the objective function influences the nature of the reorderings. In some cases this could produce some problems of interpretation. For this example, the general structure could be considered reasonably similar to the reordered matrix of Table 2 based on the sum of above diagonal entries except possibly for the Assault and Battery variable (0_4). For the particular ordering given above, out of a total of $n(n-1)(n-2)/6 = 455$ comparisons of an entry above the main diagonal to an entry below the diagonal within the same column, 454 have the above diagonal entry larger, i.e., there is only one inconsistency. Very interestingly, however, for the Table 2 matrix, based on the alternative objective function of summing the above diagonal entries, the value for this same nonmetric criterion is also 454. Thus, even though the two orderings are somewhat different, they lead to the exact same value for the nonmetric objective function. Although in most cases the fine detail of an ordering does not necessarily have a legitimate substantive importance, the matrix in Table 2 also optimizes another reasonable objective function. Thus, a researcher may have increased confidence in interpreting the specific reordering used to present the matrix in Table 2. (For completeness, we note that the ordering based on Thurstone scale values would produce an index value of 451 or 4 inconsistencies).

From a broader perspective, the ambiguity in this example suggests that any combinatorial optimization approach to matrix reorganization must be used with care if it is to be based on only one objective function. Different orderings could lead to the same optimal value and it may be problematic as to which should be interpreted as "the" result. More generally, the use of any single criterion or procedure, whether based on a Thurstone scaling model or matrix reorganization through some objective function, may provide a degree of precision that is really not possible for the fallible data being analyzed. Although the caveat may be obvious, researchers should always recognize that the choice of data analysis procedures or method of optimizing is, up to a point, arbitrary. Or stated in a different way, the ability to provide precise answers to particular data analysis questions may be more of a function of the strategy selected than of any inherent precision in the orginal data.

B. Gradients Within Rows

The general strategy of forcing large entries above the main diagonal and small entries below may be the most common strategy for reorganizing an asymmetric matrix **A** of the paired comparison form, but the underlying notion that a unidimensional seriation is being constructed suggests that gradient information may also be important. For example, suppose the n objects are assumed to fall along a single dimension and the entry a_{ij} reflects the degree to which 0_i is to the left of 0_j constructed, for example, from a paired comparison experiment. Then, besides having all the large entries above the main diagonal, there should also be a gradient within each row and column for an optimal placement. In particular, the entries within a row moving away from the main diagonal to the right should be increasing and moving toward the left away from the main diagonal the entries should be decreasing. For convenience, this latter gradient will be rephrased as an increase of the entries within a column moving up from the main diagonal.

As one possible objective function that reflects these gradient conditions, suppose $0_j \in \bar{R}$, $0_i \in \bar{R} - \{0_k\}$ and $d(R, 0_k)$ is redefined as the sum of two terms: (i) summing over all objects in R, the number of entries in $\bar{R} - \{0_k\}$ such that a_{jk} is less than a_{ji}, (ii) summing over all objects in $\bar{R} - \{0_k\}$, the number of entries in R such that a_{ki} is less than a_{ji}. Thus, the (i) term counts gradient consistencies within rows and the (ii) term counts gradient consistencies within columns. The recursion in (1) can be used with this new definition of $d(R, 0_k)$.

Again, using the Table 1 data, an ordering similar in general form to those discussed

earlier could be identified as optimal: $(0_{14}, 0_9, 0_{15}, 0_{10}, 0_1, 0_2, 0_3, 0_{13}, 0_6, 0_7, 0_5, 0_8, 0_4, 0_{11}, 0_{12})$. The value of the objective function here is 807 [out of a total of $n(n-1)(n-2)/3 = 910$ possible comparisons made]. We note that the value of the objective function for the Table 2 ordering is slightly less (a value of 791) but not dramatically so. This again reflects the overall broad consistency between the ordering given above and that used in Table 2, and in general, between the orderings that are optimal with respect to different objective functions.

C. Assuming Unfolding

If we believe that a Coombsian model is appropriate in which subjects are also placed at ideal points along the single dimension in addition to the objects, and subjects respond as a function of their distance from the objects, then a different gradient condition would be sought in **A** [see Greenberg, 1965]. The entries should increase moving either to the right or to the left of the main diagonal. Thus, for the within column consistency term (ii) given above, the phrase "less" merely has to be changed to "greater" and this new definition of $d(R, 0_k)$ used in the recursion given by (1).

D. Symmetric Matrices

The type of gradient conditions (without subject ideal points) discussed in (B) for asymmetric data apply to symmetric matrices as well. Assuming that small entries in **A** reflect similar objects, the typical unidimensional pattern would be represented by an increase within a row moving toward the right from the main diagonal and within a column moving up from the main diagonal. The relationship between these two gradients measured by the two terms given above in (B) as (i) and (ii) and the construction of *proper* interval graphs is discussed by Hubert [1974a] and Roberts [1978]. Alternatively, use of the single gradient condition (i) indexing only the degree to which the entries increase within a row moving to the right off the main diagonal would represent a natural condition for constructing approximate interval graph representations [Mirkin, 1979].

Again, considering only symmetric matrices, a number of approaches to the problem of unidimensional seriation discussed in the literature can be shown to involve combinatorial optimization components that can be attacked with the type of recursion given in (1). For example, Defays [1978] seeks coordinates $x_1, x_2, \ldots, x_n$ for the n objects $0_1, 0_2, \ldots, 0_n$, such that

$$\sum_{i<j} (a_{ij} - |x_i - x_j|)^2$$

is minimized. Defays' solution requires an initial reorganization of **A** to maximize an objective function defined by taking the sum of the squared differences between the sum of the entries within each row to the left of the main diagonal and the sum of the entries within the corresponding column up from the main diagonal. The recursion in (1) is appropriate with $d(R, 0_k)$ defined as the square of the difference between the sum of proximities from 0_k to $\bar{R} - \{0_k\}$ and from R to 0_k.

E. Szczotka's Criterion

A second objective function discussed by Szczotka [1972], among others, is relevant to the placement of the objects one unit apart along a continuum and is defined by

$$\sum_{i<j} a_{\rho(i)\rho(j)} |i - j|.$$

Since small values of this objective function are desirable, the same type of recursion as in (1) is appropriate but in a minimization form with $d(R, 0_k)$ given by $m + 1$ times the

difference between the sum of proximities from R to 0_k and from 0_k to $\bar{R} - \{0_k\}$ [see Elmaghraby, 1968, for a more complete discussion and justification for this definition of $d(R, 0_k)$].

Discussion

There are several other dynamic programming approaches to matrix reorganization that could be mentioned for completeness. First, if the rows and columns of **A** are to be relabeled in a way that produces large entries immediately adjacent to the main diagonal (i.e., the entries of the form $a_{i(i+1)}$, $1 \leq i \leq n-1$), then the same type of dynamic programming recursion could be used that works for the well-known traveling salesman problem [Reinhold, Nievergelt, & Deo, 1977]. Various data analysis implications of this optimization task are discussed in detail by Hubert and Baker [1978]. Another dynamic programming strategy discussed by Adelson, Norman and Laporte [1976] is appropriate for reordering the rows of a rectangular matrix containing zeros and ones; here, our aim is to produce an approximate sequence of consecutive ones in each column. The data analysis uses of this task are presented by Hubert [1974b]. In these two cases as well as the examples we have given for the recursion in (1), the general aim is to reduce the burden of evaluating some objective function over a set of permutations to the smaller task of finding the longest path in a network of nodes defined by all possible subsets of the objects being studied. Even though this latter problem is still computationally difficult, the availability of a dynamic programming strategy at least extends the range of matrix sizes for which optimal solutions can be obtained.

As a somewhat more general context for the type of matrix reorganization strategies we have proposed, it is possible to reformulate most of these optimization tasks in terms of the quadratic and cubic assignment problems. The term *quadratic assignment* (QA), for example, refers to a very general combinatorial optimization task with an extensive literature in the area of operations research and management science (for example, see Hanan & Kurtzberg, 1972). In its most typical multiplicative form, two $n \times n$ matrices $\{c_{ij}\}$ and $\{a_{ij}\}$ are given, both with zeros along their main diagonals, and we attempt to permute rows, and simultaneously columns, of **A** so as to maximize (or minimize)

$$\Gamma(\rho) = \sum_{i,j} a_{\rho(i)\rho(j)} c_{ij} ,$$

where $\rho(\cdot)$ is permutation of the first n integers. For example, $\{c_{ij}\}$ may be considered a matrix of numerical relationships between n facilities, where the entry c_{ij} reflects the amount of material to be moved from facility i to facility j. In turn, the matrix a_{ij} could be viewed as a second matrix containing distances among n positions or locations. Each of the n facilities could be placed at any of the n locations. Since $\rho(\cdot)$ represents an assignment of facilities to locations (i.e., if $\rho(u) = v$, then facility u is assigned to location v), the QA optimization task is equivalent to finding an assignment that minimizes a total effort or cost represented by the index $\Gamma(\cdot)$.

It has been recognized for some time that QA includes a number of difficult combinatorial optimization tasks as special cases. For example, the variation on the traveling salesman problem just mentioned in which the sum of entries immediately adjacent to the main diagonal is to be maximized can be characterized as a maximization of $\Gamma(\cdot)$ with $c_{ij} = 1$ if $j = i + 1$ and 0 otherwise [Lenstra & Rinnooy Kan, 1975]. In a similar way, the maximization of the above diagonal entries used in our initial example is defined by $c_{ij} = 1$ if $i < j$ and 0 otherwise [Blin, 1973] and the linear seriation method discussed by Szczotka [1972] in (*E*) is obtained for $c_{ij} = |i - j|$.

Given the form of the QA index, a generalization to cubic assignment (CA) is straightfor-

ward. Beginning with two 3-place functions q_{ijk} and p_{ijk}, the CA optimization task is to locate a permutation $\rho(\cdot)$ that minimizes

$$\Lambda(\rho) = \sum_{i,j,k} q_{\rho(i)\rho(j)\rho(k)} p_{ijk}.$$

As one special case of interest here, it is possible to consider a very broad class of CA problems by defining the two functions q_{ijk} and p_{ijk} in very general terms from the corresponding matrices that would be used in QA framework. For example, one natural class of indices could be based on a triads measure suggested by Hartigan [1975] in the context of comparing two hierarchical clustering structures. Following Hartigan's lead and based on the two matrices $\{c_{ij}\}$ and $\{a_{ij}\}$ used for the QA index, q_{ijk} and p_{ijk} would be defined as:

$$q_{ijk} = \begin{cases} 1 & \text{if } i, j, k \text{ are distinct and } a_{ik} < a_{jk}; \\ 0 & \text{otherwise.} \end{cases}$$

$$p_{ijk} = \begin{cases} 1 & \text{if } i, j, k \text{ are distinct and } c_{ik} < c_{jk}; \\ 0 & \text{otherwise} \end{cases}$$

In words, $\Lambda(\cdot)$ counts the number of consistent triads between two matrices, where a consistent triad refers to three distinct objects 0_i, 0_j and 0_k, with the property that $a_{ik} < a_{jk}$ and $c_{ik} < c_{jk}$. Clearly, only the rank orders of the values in the two matrices $\{a_{ij}\}$ and $\{c_{ij}\}$ are relevant when $\Lambda(\cdot)$ is defined from the functions given above. This latter property is particularly noteworthy since the sole reliance on proximity rank order that is possible in CA has been a prime motivating concern for almost two decades in the psychological literature, e.g., in cluster analysis [Johnson, 1967] and multidimensional scaling [Shepard, 1962].

Although it may be obvious that only the proximity rank orderings are needed when the two functions q_{ijk} and p_{ijk} are given by the definitions above, this invariance argument can be carried one step further. Specifically, since q_{ijk} and p_{ijk} are "keyed" on the last subscripts, comparisons are made only within the columns of $\{a_{ij}\}$ and $\{c_{ij}\}$. This implies that the minimization of $\Lambda(\cdot)$ is unaffected when different "metrics" are used in obtaining the columns of either $\{a_{ij}\}$ or $\{c_{ij}\}$. Matrices of this latter type are very common in the behavioral sciences and are called *conditional* proximity matrices, among other names [Shepard, 1972].

To illustrate the use of this triads measures, suppose c_{ij} is defined as 1 when $i < j$, i.e., in the QA paradigm we would seek to maximize the sum of above diagonal entries. In the triad context, however, we now count the number of times an entry above the main diagonal is greater than an entry below the main diagonal within the same column. This is the objective function considered earlier in (B).

If we define the triad measures differently, the CA task also includes the gradient objective function discussed in (B) and (D). For instance, let $q_{ijk} = 1$ if i, j and k are distinct and either (i) $a_{ki} < a_{kj}$ for $k < i$ and $k < j$ or (ii) $a_{ik} < a_{jk}$ for $i < k$ and $j < k$, q_{ijk} is zero otherwise. If p_{ijk} is defined similarly when $c_{ij} = |i - j|$, then $\Lambda(\cdot)$ defines the objective functions used in (B) and (D). The counts referenced here within the rows and columns, respectively, as (i) and (ii), correspond directly to the same distinction in (B). Thus, as in (C), the use of only one of these gradient conditions, i.e., (i) or (ii), could also be considered.

REFERENCES

Adelson, R. M., Norman, J. M., & Laporte, G. A dynamic programming formulation with diverse applications. *Operational Research Quarterly*, 1976, *27*, 119–121.

Arabie, P. & Carroll, J. D. MAPCLUS: A mathematical programming approach to fitting the ADCLUS model. *Psychometrika*, 1980, *45*, 211–235.

Baker, F. B. & Hubert, L. J. Applications of combinatorial programming to data analysis: Seriation using asymmetric proximity measures. *British Journal of Mathematical and Statistical Psychology*, 1977, *30*, 154–164.

Blin, J. M. On tournament patterns and rational group decisions. *Proceedings of IEEE 1973 Conference on Cybernetics and Society*, 1973, 55–59.

Defays, D. A short note on a method of seriation. *British Journal of Mathematical and Statistical Psychology*, 1978, *31*, 49–53.

Dreyfus, S. E. An appraisal of some shortest-path algorithms. *Operation Research*, 1969, *17*, 395–412.

Elmaghraby, S. E. The sequencing of "related" jobs. *Naval Research Logistics Quarterly*, 1968, *15*, 23–32.

Flueck, J. A. & Korsh, J. F. A branch search algorithm for maximum likelihood paired comparison ranking. *Biometrika*, 1974, *61*, 621–626.

Greenberg, M. G. A method of successive cumulations for scaling of pair-comparison judgements. *Psychometrika*, 1965, *30*, 441–448.

Hanan, M. & Kurtzberg, J. M. Placement techniques. In M. A. Breuer (Ed.), *Design Automation of Digital Systems*. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

Hartigan, J. A. *Clustering Algorithms*. New York: Wiley, 1975.

Holman, E. W. Completely nonmetric multidimensional scaling. *Journal of Mathematical Psychology*, 1978, *18*, 39–51.

Hubert, L. J. Some applications of graph theory and related nonmetric techniques to problems of approximate seriation: The case of symmetric proximity measures. *British Journal of Mathematical and Statistical Psychology*, 1974, *27*, 133–153. (a)

Hubert, L. J. Problems of seriation using a subject by item response matrix. *Psychological Bulletin*, 1974, *81*, 976–983. (b)

Hubert, L. J. Seriation using asymmetric proximity measures. *British Journal of Mathematical and Statistical Psychology*, 1976, *29*, 32–52.

Hubert, L. J. & Baker, F. B. Applications of combinatorial programming to data analysis: The traveling salesman and related problems. *Psychometrika*, 1978, *43*, 81–91.

Johnson, S. C. Hierarchical clustering schemes. *Psychometrika*, 1967, *32*, 241–254.

Karp, R. M. Reducibility among combinatorial problems. In R. E. Miller & J. W. Thatcher (Eds.), *Complexity of Computer Computations*, New York: Plenum Press, 1972, 85–104.

Korte, B. & Oberhofer, W. Triangularizing input-output matrices and the structures of production. *European Economic Review*, 1971, *2*, 493–522.

Lawler, E. L. A comment on minimum feedback arc sets. *IEEE Transactions on Circuit Theory*, 1964, *11*, 296–297.

Lenstra, J. K. & Rinnooy Kan, A. H. G. Some simple applications of the traveling salesman problem. *Operational Research Quarterly*, 1975, *26*, 717–733.

Mirkin, B. G. *Group choice*. New York: Wiley, 1979.

Reinhold, E. M., Nievergelt, J. & Deo, N. *Combinatorial algorithms: Theory and practice*. Englewood Cliffs, N. J.: Prentice-Hall, 1977.

Roberts, F. S. *Graph theory and its applications to problems of society*. Philadelphia: Society for Industrial and Applied Mathemetics, 1978.

Shepard, R. N. The analysis of proximities: Multidimensional scaling with an unknown distance function I, II. *Psychometrika*, 1962, *17*, 125–140, 219-246.

Shepard, R. N. A taxonomy of some principal types of data and of multidimensional methods for their analysis. In R. N. Shepard *et al.* (Eds.), *Multidimensional scaling: Theory and applications in the behavioral sciences* I. New York: Seminar Press, 1972.

Szczotka, F. A. On a method of ordering and clustering of objects. *Zastosowania Mathemetyki*, 1972, *13*, 23–33.

Thurstone, L. L. The method of paired comparisons for social values. *Journal of Abnormal and Social Psychology*, 1927, *31*, 384–400.

Manuscript received 2/2/81
Final version received 7/28/81