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., $\rho: (1, 2, \ldots, n) \rightarrow (1, 2, \ldots, n)$, 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 $n!$
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 $15 \times 15$. 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 $A$.  