RICE UNIVERSITY STUDIES
JOURNAL OF WILLIAM MARSH RICE UNIVERSITY

Rice University Studies, successor to the Rice Institute Pamphlet, is a quarterly journal of writings in all scholarly disciplines by staff members and other persons associated with Rice University.

EDITOR: Katherine Fischer Drew
ASSISTANT EDITOR: Mimi G. Cohen

RICE UNIVERSITY STUDIES COMMITTEE
K. F. Drew, Chairman, J. W. Campbell, Richard O'Neil,
Madeleine R. Raaphorst, Carroll Camden

INFORMATION FOR AUTHORS
Manuscripts submitted for publication should be addressed to the Editor, Rice University Studies, Rice University, Houston, Texas. When preparing manuscripts contributors in the humanities and social sciences are requested to follow the MLA Style Sheet; contributors in science and mathematics are requested to follow the established procedures of major journals in their special fields. Notes and references will appear at the end of the manuscript.

Individual numbers of Rice University Studies are available at $2.25 plus cost of postage and handling from Rice Campus Store, P. O. Box 1892, Houston, Texas 77001.

Second class postage paid at Houston, Texas.
Acknowledgments

Various parts of the research reported here were supported by different agencies. The archeological field work from which the original data used in this study were derived was supported at various times by Rice University, the University of Chicago, and the National Science Foundation in grants GS 67 and GS 724 made to Frank Hole. Subsequent study of these data was supported by the same agencies and the Committee on Applied Mathematics and Systems Research, Rice University. Computer time to develop PHOENIX-I was granted by Rice University, and Carnegie-Mellon University provided facilities for the development of PHOENIX-II under Contract SD-146 with the Advanced Research Projects Agency of The Office of the Secretary of Defense.

We should especially like to thank the following persons for their advice or help on particular aspects of this project: Keith A. Dixon, Carl L. Lefkowitz, Gary Sitton, and Roy R. Weil.

FRANK HOLE
Associate Professor of Anthropology
RIE UNIVERSITY

MARY SHAW
Department of Computer Science
CARNegie-MELLON UNIVERSITY
# Computer Analysis of Chronological Seriation

## I. Dating by Seriation
- Introduction ........................................... 1
- Dating ............................................... 1
- Theories of Seriation ................................. 3
- Methods of Seriation ................................ 4
- Typology ............................................. 5
- Archeological Data ................................. 6

## II. Techniques for Automatic Seriation
- The Brainerd-Robinson Technique .................. 8
- The Meighan Technique .............................. 11
- The Dempsey-Baumhoff Technique ................. 11
  - Presence-Absence Matrices ....................... 12
  - Contextual Analysis ............................. 13
- The Permutation Search Technique ............... 13
  - Details of the Search Technique ................ 15
  - Input Order ..................................... 21
  - Constraints .................................... 23
- The Type-Percentage Technique ................... 24

## III. Evaluating Results
- Matrix Norms ...................................... 27
  - Proposals for Norms ............................. 27
  - Our Choice of Norm ............................. 28
  - Comparing Different Matrices .................. 29
- Computation Time .................................. 31
- Error Coefficient .................................. 34
- Characteristics of Data ........................... 36
  - Recognizing Good and Bad Data ................ 37
  - Improving Poor Data Sets ....................... 38
- Display of Results ................................ 39
  - Contour Plots .................................. 40
  - Percentage Plots ................................ 43

## IV. Comparison of Results
- The Illustrative Data ............................... 51
  - The Deh Luran Sequence (Data Sets I, II, IV) 51
  - The Khorramabad Sequence (Data Set III) .... 53
  - Seriation of Data Sets .......................... 53
| Data Set I. Deh Luran Pottery                      | 54 |
| Data Set II. Deh Luran Flint                     | 59 |
| Data Set III. Pa Sangar Flint                    | 62 |
| Data Set IV. 110 Types                           | 64 |

**V. EVALUATION OF TECHNIQUES**

| The Brainerd-Robinson Approach                  | 67 |
| The Ascher Technique                            | 68 |
| The Meighan Technique                           | 69 |
| The Dempsey-Baumhoff Technique                  | 77 |
| Presence-Absence Coefficients                   | 78 |
| Contextual Analysis                             | 78 |
| The Permutation Search Technique                | 80 |
| The Type-Percentage Technique                   | 82 |

**VI. CONCLUDING REMARKS**

| Theoretical-Cultural Considerations             | 84 |
| Archeological Considerations                    | 85 |
| Mechanical Considerations                       | 87 |
| General System Design Standards                 | 89 |
| PHOENIX: A System for Automatic Seriation       | 91 |
| The Permutation Search Technique: A Planned Program | 94 |
| Open Questions and Suggestions for Future Study | 94 |

**APPENDIX A. TABLES OF RAW DATA**                     | 97 |

**APPENDIX B. COMPLETE OUTPUT FOR DATA SUBSET Ia**     | 111 |

**APPENDIX C. USER'S GUIDE TO PHOENIX-II**             | 119 |

**APPENDIX D. PROGRAM LISTING FOR PHOENIX-II**         | 135 |

**APPENDIX E. PROGRAMMING NOTES ON PHOENIX-II**       | 157 |

**BIBLIOGRAPHY**                                       | 167 |
# LIST OF TABLES

2-1. Cost of Evaluating Permutations During the Ordering of a Matrix ........................................ 14

3-1. Frequency Distribution of Values in Correlation Matrices .................................................. 30

3-2. Cost of Computing Various Matrix Norms ................................................................. 32

4-1. The Deh Luran Stratigraphic Sequence .............................................................................. 52

4-2a. Orderings for Data Set I, Deh Luran Pottery (Brainerd-Robinson Matrix) ......................... 54

4-2b. Orderings for Data Set I, Deh Luran Pottery (Presence-Absence Matrix) ......................... 54

4-3a. Orderings for Data Subset Ia, Susiana Black-on-Buff Pottery (Brainerd-Robinson Matrix) .................................................. 55

4-3b. Orderings for Data Subset Ia, Susiana Black-on-Buff Pottery (Presence-Absence Matrix) .................................................. 55

4-4a. Orderings for Data Subset Ib, Susiana Buff Pottery (Brainerd-Robinson Matrix) ............. 57

4-4b. Orderings for Data Subset Ib, Susiana Buff Pottery (Presence-Absence Matrix) ............. 57

4-5a. Orderings for Data Subset Ic, Bayat Red Pottery (Brainerd-Robinson Matrix) ................ 58

4-5b. Orderings for Data Subset Ic, Bayat Red Pottery (Presence-Absence Matrix) ................ 58

4-6a. Orderings for Data Set II, Ali Kosh-Tepe Sabz Flint (Brainerd-Robinson Matrix) .......... 60

4-6b. Orderings for Data Set II, Ali Kosh-Tepe Sabz Flint (Presence-Absence Matrix) .......... 60

4-7. Orderings for Data Set III, Pa Sangar Flint, 33 Types .................................................. 61

4-8a. Orderings for Data Subset III, Pa Sangar Flint, 11 Types (Brainerd-Robinson Matrix) ........ 63

4-8b. Orderings for Data Subset III, Pa Sangar Flint, 11 Types (Presence-Absence Matrix) ........ 64

4-9a. Orderings for Data Set IV, 110 Types (Brainerd-Robinson Matrix) .................................. 65

4-9b. Orderings for Data Set IV, 110 Types (Presence-Absence Matrix) .................................. 66

5-1. Data for 3-Pole Plot of Deh Luran Pottery, Types 23, 25 (Susiana Black-on-Buff), Type 42 (Susiana Buff) .................................................. 71

5-2. Data for 3-Pole Plot of Deh Luran Pottery, Types 15, 23, 25 (Susiana Black-on-Buff) .......................................................................................... 72

5-3. Data for 3-Pole Plot of Pa Sangar Flint, Types 18, 26, 36 .................................................. 74

5-4. Comparison of Results for Dempsey-Baumhoff Example ................................................. 77

5-5. Data Subset Ia, Pattern III Artifacts Removed .............................................................. 80

5-6. Data Subset Ib, Pattern III Artifacts Removed .................................................................. 80
LIST OF FIGURES

2-1. Example of permutation search ........................................... 18, 19
2-2. Contour plot of correlation matrix in correct order
    (Robinson’s data) ............................................................. 22
2-3. Contour plot of correlation matrix in incorrect order
    (Robinson’s data) ............................................................. 23
3-1. Histogram of frequency distribution of values in
    correlation matrices ......................................................... 31
3-2. Plot of cost of computing various matrix norms ...................... 33
3-3. Percentage distribution for three ideal types ......................... 36
3-4. Meighan 3-pole plot of three ideal types ............................. 37
3-5. Contour plot of a perfect sequence of 20 sites with uniform
    spacing of intervals ......................................................... 39
3-6. Contour plot of a matrix from the data used in Figure 3-5
    (with sites numbered 5, 9, 10, 14, 15, and 16 removed) .......... 40
3-7. The best order found for Data Set I ................................... 41
3-8. Contour plot of data used in Figure 3-5
    (data in a scrambled order) .............................................. 42
3-9. Contour plot of correlation matrix in incorrect order
    (Robinson’s data in input order) ....................................... 43
3-10. Contour plot of Data Subset Ia (Susiana Black-on-Buff pottery) .. 44
3-11. Contour plot of Data Set I, (Deh Luran pottery) 10 contours .... 44
3-12. Contour plot of Data Set I, (Deh Luran pottery) 8 contours .... 45
3-13. Contour plot of Data Set I, (Deh Luran pottery) 6 contours .... 45
3-14. Contour plot illustrating technique of making width of lines
    proportional to slope ...................................................... 46
3-15. Contour plot illustrating technique of making lines of
    equal width ...................................................................... 46
3-16. Photograph of contour plot produced on an oscilloscope
    (Robinson’s data) ............................................................. 47
3-17. Photograph of stereographic contour plot produced on an
    oscilloscope (Robinson’s data) .......................................... 47
3-18. Photograph of contour plot produced on an oscilloscope
    (Susiana Black-on-Buff pottery) ....................................... 48
3-19. Photograph of stereographic contour plot produced
    on an oscilloscope (Susiana Black-on-Buff pottery) ............. 48
3-20. Cross sections through typical contour plot to show
    characteristics of topography ........................................... 49
3-21. Type-Percentage plot of Robinson’s data .......................... 50
4-1. Contour plot of Data Set III (Pa Sangar flint),
    components grouped by fours ........................................... 62
5-1. Ordering a correlation matrix by the Ascher method
    (data hypothetical) ....................................................... 70
5-2. 3-pole plot of types in Table 5-1 ........................................ 72
5-3. 3-pole plot of types in Table 5-2 ........................................ 73
5-4. 3-pole plot of types in Table 5-3 ........................................ 74
5-5. Permutation Search order plotted on 3-pole graph .................. 75
C-1. Input form for a TASK which reads and orders raw data .......... 130
C-2. Three TASKs which order published data ............................ 132
C-3. Input for a complete PHOENIX run .................................. 133
CHAPTER I
DATING BY SERIATION

Introduction

Any reconstruction of the history of a culture requires that the events be placed in a chronological order. In view of this fundamental importance of dating, archeologists have tested and pioneered in the development of many techniques. At the present, the dating of archeological remains can be accomplished by a number of independent methods which yield chronologies of varying degrees of precision. Unfortunately, most of these techniques are limited in their application and the deposits in a typical archeological site may, therefore, need to be dated by several methods. It is toward a refinement of dating by one such method—the seriation of artifacts—that we direct our attention in this study.

Techniques to deduce chronological order by analyzing systematic changes in assemblages of artifacts are important tools that can be used independently or in conjunction with other dating methods. Intuition had long suggested, and specific studies had demonstrated, that changes in artifacts may be linearly related to time. Again, intuition suggested, but it had not been systematically tested, that some archeological data are more amenable to seriation than others. In fact, a number of seriations and other chronological orderings of archeological material had been performed by archeologists over the years with little concern as to the suitability of the techniques or the validity of the theories on which they were based.

It was with these issues in mind that we undertook the present project. Without the use of computers, any large-scale testing program would have been impossible. It was therefore natural that the evaluation and development of methods of seriation went hand in hand with the more general theoretical aspects of this study.

The project described in this monograph is an evaluation and refinement of the theories and methods that have been proposed concerning the chronological seriation of artifacts. There are many examples in the archeological literature of the use of seriation for relative dating as if there were no question of its worth (e.g., Ford 1962, Kuzara, Mead and Dixon 1966, and Willey 1953). Studies that have questioned aspects of seriation include Dixon's work (1956) with the Snaketown artifacts and Deetz and Dethlefsen's study (1965) of grave markers in Massachusetts.
The theory behind seriation is clear, and in many cases is known to be correct, but it has never been tested systematically with data of known order. In view of its potential importance and because we had abundant data of various kinds in the proper form for seriation we decided to carry out extensive tests of the method. The mechanical operations were performed by computers at Rice University and Carnegie-Mellon University.

We reasoned that if methods of seriation are generally applicable, they should arrange our archeological components in what we know to be their stratigraphic positions. At a minimum we would thus be able to test methods of seriation. On the other hand, if our data did not seriate into their known stratigraphic order we should have to decide among the following possibilities: (1) the stratigraphy did not distinguish the true chronology; (2) the assumptions on which we based our ordering and the techniques we used were incorrect; (3) the data were not suitable.

In our studies we found that we can seriate some of our data so that they duplicate the stratigraphic order, and most of our data so that they approximate the stratigraphic order, but we have also found that not all data are amenable to seriation. As a consequence, we learned a number of things about methods for deriving chronology from assemblages of artifacts and we learned some things about our particular data. It is to these theoretical and methodological considerations of seriation that we address ourselves in this paper.

In recent years some of the techniques of seriation have been expressed as computer programs to allow machines to do the difficult mechanical part of the data manipulation and to make feasible large-scale comparisons (e.g., Ascher and Ascher 1963, and Kuzara, Mead and Dixon 1966). These computer programs and some of the other procedures that have been proposed are called "algorithms." As we use the term algorithm, it denotes a procedure which is expressed in such a way that at no point is there any question about what action to perform and what step to take next; the procedure must, moreover, terminate in a finite length of time. In essence an algorithm is an explicit, unambiguous procedure for carrying out a computation.

We have chosen several of these algorithms, and by comparing their performance on our data we investigated the methods of seriation, the relation between data and results, and the theory of systematic change in artifacts.

The contributions of this project lie in the following areas:
1. An evaluation of the theoretical bases for understanding systematic changes in artifacts.
3. An investigation of the performance of particular archeological data in cases of seriation.
4. The development of a system of computer programs capable of seri-
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION

ating archeological data accurately and obtaining at least qualitative evaluations of dependability.

The computer system developed in the course of this study includes a new algorithm for seriation called Permutation Search. It is described in Chapters II and V and in Appendices B, C, and D. Our evaluations of, and conclusions on, the performance of other algorithms for seriation are given in Chapters II and V. Our more general conclusions and remarks on seriation and seriation techniques are contained in Chapter VI. In Chapter IV we illustrate seriation by showing the results of applying various algorithms to separate sets of data.

**Dating**

The concept of dating itself requires some explanation because in various contexts it may mean different things. In popular language, dates indicate the number of years that have elapsed since a particular event (e.g., the years since the birth of Christ), but archeologists use the term “date” in other senses too. Archeologists make a basic conceptual distinction between absolute (calendar) and relative dates. Of course, all calendar dates are relative to the starting point of the calendar, but they can be expressed in absolute terms too. On the other hand, relative dates can only be expressed as “older than” (or “more recent than”) other events or dates. When we refer to dates or to chronology in this paper we shall mean relative dates: that site or event A is older than B, and B is older than C.

Several methods, most notably radiocarbon dating, give what seem to be absolute dates: “The C-14 date of this site is 3750 ± 200 B.C.” What is often ignored is that a radiocarbon date is a numerical expression of the probability that the sample of carbon being dated falls within a range of possible dates. The date is absolute in the sense that it is expressed in calendar years but it is only approximate in the sense that it has a very low probability of being precisely the age given and only two out of three chances of being within one standard deviation. In many archeological situations, radiocarbon dating does not give dates precise enough to permit an accurate relative chronology of all the deposits. In these cases it is important to have an independent system of dating that will place material relatively.

Another aspect of relative dating that should be made clear is that there is no absolute duration for a unit of archeological time. A calendar year is approximately 365.25 days long, but archeological time can only be measured in terms of things that can be seen changing; in the cases we describe here, artifacts. We must think of the discernible changes in artifacts as units of time rather than thinking of the days or years implied by the changes. The scale of discernible chronology will thus refer to varying absolute lengths of time which will remain unknown in the absence of a
method of absolute dating. For example, archeological period A may have a duration of 100 years, period B of 200 years, and period C of 50 years. Archeologically they may appear to be equal time units and often our techniques of discerning chronology are not sufficient to allow more than a guess as to the absolute duration of each period; all we are able to do is to put the periods in their relative chronological order.

Theories of Seriation

Methods for chronological ordering are based on the theory that during any period of time people make and use a unique assemblage of artifacts. This theory is usually stated as follows: "Each type [of artifact] originates at a given time at a given place, is made in gradually increasing numbers as time goes on, then decreases in popularity until it becomes forgotten, never to recur in an identical form" (Brainerd 1951: 304).

It follows that each period is characterized both by its unique assemblage of artifacts and by the relative frequency with which each type of artifact occurs. So far as we are aware, no one has refuted the general validity of these basic principles, although in particular examples they may be incorrect. Hence, presence or absence and relative abundance of artifacts are two tools with which one can work in seriation. In consequence, sites showing the greatest degree of agreement either in the occurrence of artifacts or in their frequencies will lie closest in time. One of the values of the present study has been to show some instances where one or both of the premises does not hold and to set up some theoretical ground rules to follow when attempting seriation.

Methods of Seriation

"It looks old," is the criterion most people use for ordering things they see. Archeologists are only relatively more sophisticated in their ability to recognize things of different ages; it is commonplace for an archeologist to be able to tell that artifact W is younger than X but older than Y. We base these judgments on the attributes of artifacts just as boys do when they can tell the relative ages of cars. A classic example of this sort of observation was made by Petrie (1899) who, after examining the contents of hundreds of Egyptian graves, was able to seriate the pottery chronologically by merely looking at the characteristics of the handles. Rowe (1959) describes a recent use of the Petrie method as it is applied to Athenian pottery where skilled archeologists can date pots to within 25-30 years by recognizing minute features in their construction and decoration. Archeologists with experience in an area can make similar judgments about certain kinds of artifacts; these are usually called "horizon markers" or "index fossils" (for uses of this technique with pottery, see Adams 1965; with carved stones, see Proskouriakoff 1950, and with grave stones, see Deetz and Dethlefsen 1965). It should be evident, however, that only after artifacts and art
styles have been dated independently can they be used themselves as a means for dating associated artifacts or features.

One might routinely depend on index fossils except that many artifacts change too slowly to yield results that are precise enough, and others that are good horizon markers are too rare to be found frequently. Therefore, other methods had to be developed to date collections which were known, or thought, to be of different ages. The most important of these, using numerical methods of chronological ordering, are reviewed in Chapter II.

**Typology**

Any system of seriation that depends on changes in types of artifacts requires careful definition and explanation of the types. American archeologists use the concept of “type” routinely, although not all of them mean the same thing by the term. For the present paper we offer the following explanation. Artifacts are characterized by their attributes: such things as color, shape, size, decoration, details of chipping, etc., are examples of attributes. When certain attributes occur repeatedly in association, we regard all artifacts having these characteristics as members of a single type.

Types should be defined in such a way that they have mutually exclusive clusters of attributes or else the typology will reflect a measure of the analyst’s subjectivity. This is a particular problem when the attributes are segments of a continuous metric dimension. In these cases the isolation of types is arbitrary and, owing to imprecision of measurement, may be confused by the analyst’s inability to consistently distinguish two types. This is also the case when some of the attributes are impressionistic qualities (see Lipe 1964: 103-4, and Ford 1962: 44).

Types are tools to be used in analysis; therefore it is immaterial whether or not they would have been distinguished by the people who made them. In the initial classification it is well to subdivide rather than to lump possible types so that minor differences in style or technique (which may have chronological implications) will be separated.

It is also important for the types to be roughly equivalent in chronological importance and one ordinarily assumes that they are. However, if one item is important and two are not, the unimportant items (especially if popular) may obscure the chronological implications of the good one. This is a problem of weighting which is also encountered if the types are not mutually independent or if they are functionally related. In either case, if the two related items are counted separately, it has the effect of weighting one single thing twice as heavily as each of the other things. In this instance, two types do not tell any more about chronology than one type, and one factor may be weighted inadvertently at the expense of others. Although we recognize this problem, we have not found a way to solve it.

In seriation we need types that behave in predictable ways with respect
to time, but when we define our types we may not know how they vary chronologically. This knowledge may not come until after seriation or subsequent stratigraphic excavations have revealed the proper time sequence. It is true that the more consistently archeologists manage to define types by attributes that have chronological significance, the sharper will be the resulting seriation. However, as Brainerd (1951:303) points out, even poor typology—if objectively arrived at—should not greatly affect the outlines of the final ordering.

For reasons of economy we have not included the definitions of the types we use here. These details can be obtained by referring to Hole, Flannery and Neely (in press), and Hole and Flannery (unpublished).

**Archeological Data**

Methods that compare the presence or absence of types or the frequency distribution of types among archeological components require samples of data that are true representations of the contents of sites because any differences are automatically regarded by these methods as being chronologically significant. Faulty samples will therefore lead to incorrect conclusions. There is a literature on extracting statistically valid samples from some kinds of archeological sites but there has been all too little consideration of the problems of sampling the complex, multi-component sites we find in Southwest Asia.

Clearly a number of serious theoretical and practical problems are raised here and we can also cite some other potential sources of error. As Freeman and Brown (1964:126) explain, if separate areas of a site were used for different purposes, samples of artifacts taken from each area may reflect functional differences but be interpreted in seriation as temporal differences. We recognize these problems but we ignore them in the present analysis for the following reasons. In all instances except one, we know the stratigraphic order of our components. This gives us an independent means of interpreting our seriations and eliminates the possibility that we might be systematically mistaking functional for temporal differences. Furthermore, since our excavations amounted to test pits into what we judged were essentially homogeneous deposits (principally middens), we have no obvious way to isolate functionally independent clusters of artifacts.

The question of the nature of the samples of data used in analysis is obviously of more than passing theoretical importance and it will certainly have to be dealt with in future excavations. Since we could not re-enter the excavations to gather the data on a different basis, we preferred at the beginning of this study to assume that the data we used were comparable and that our decisions to subdivide the series into components based on natural stratigraphy were correct. This seemed a reasonable course to take in the present paper because we are as much interested here in dis-
covering how different algorithms handle a given set of data as we are in the conclusions we may be able to reach about the data. As the project proceeded, however, we were able to make judgments concerning our data and these in turn led to a better appreciation of some of the theoretical aspects of seriation.
CHAPTER II

TECHNIQUES FOR AUTOMATIC SERIATION

Most archeologists follow the method outlined by Ford (1962) when they seriate their deposits. Ford’s booklet describes a manual process of seriation which is time-consuming, tedious, and frequently inaccurate. To make the process easier and less subjective, several methods for performing the seriation tasks have subsequently been proposed. In essence, automatic seriation means that numerical indices of similarity replace visual and other subjective judgments. The present study considers five different approaches to automatic seriation: the Brainerd-Robinson, Meighan, and Dempsey-Baumhoff methods, and two new methods, which were developed in the course of this study, Permutation Search and Type-Percentage.

The Brainerd-Robinson Technique

This technique, first proposed by Robinson (1951), a statistician, in response to a problem posed by Brainerd (1951), an archeologist, attempts to order sites on the basis of a numerical index of the degree to which two sites are similar. A correlation or agreement matrix is constructed by comparing the percentage distribution of artifacts in each possible pair of sites. The seriation technique then consists of finding an order in which the numbers in the correlation matrix fall into a certain pattern. Brainerd and Robinson did not provide a technique other than reasoned trial-and-error for finding the internal pattern they described; this was attempted by Ascher and Ascher (1963), who programmed an explicit procedure for a digital computer.

Because we use the Brainerd-Robinson matrix as one form in which to present our data it is useful to thoroughly review its characteristics, even though these details are given in the original articles.

Robinson based his seriation on ordering rows and columns of a correlation matrix. The first step of a procedure to accomplish this was to define an agreement coefficient, a numerical expression of the differences (disagreement) between two sites. If two sites do not have identical distributions of artifacts, some or all of the types will occur with different percentages in the two sites. The disagreement of the sites may be expressed as the sum of these discrepancies in type percentages. Since the maximum value of disagreement is thus 200 when there is no type which is found at
both sites, and the minimum disagreement is 0 when the sites are identical, the agreement of the sites (the index generally used) may be measured by subtracting the disagreement from 200. The agreement coefficient will thus range from 0 (no agreement) to 200 (perfect agreement).

If we let

- \( P \) represent the number of sites
- \( T \) represent the number of types
- \( A_{I,J} \) represent the percentage of site J composed of type I
- \( X_{I,J} \) represent the correlation between site I and site J

then the Brainerd-Robinson agreement coefficient may be formally defined:

\[
X_{I,J} = 200.0 - \sum_{K=1}^{T} |A_{K,I} - A_{K,J}|
\]

for \( I, J = 1, 2, \ldots, P \).

Next, Robinson constructed the matrix by computing agreement coefficients between each of the possible pairs of sites, entering the coefficient expressing the agreement of site J with site K at the intersection of column J with row K. Since each site correlates perfectly with itself, the diagonal elements will all be 200. It follows that the agreement between J and K must be the same as that between K and J, making the correlation matrix symmetric; that is, the halves of the matrix on either side of the line of perfect agreement will be mirror images (e.g., see Appendix B).

According to Robinson’s theory, sites closest in time have the highest agreement coefficients. The agreement coefficients will diminish as sites become farther removed in time. He went on to reason that “if the deposits are chronologically arranged along the margins . . . the resultant pattern of agreement indexes will show a definite structure, in that as any row is read from left to right the indexes will progressively grow larger up to the diagonal, and thence will progressively decline from that point on . . . In other words, if the deposits are chronologically arranged along the margins, the resulting table of agreement indexes will show high values clustering about the diagonal, with decreasing values as one goes away from the diagonal either vertically or horizontally” (Robinson 1951:294-295).

Experience in working with archeological data shows that a correlation matrix almost never fits perfectly into this ideal form; therefore, the user of a seriation method must attempt to find the ordering of sites that gives the best approximation to an ideal matrix pattern rather than always to expect a perfect fit.

In his analysis of stratified sites, Brainerd found a tendency for the “matrix to slope upward toward one end of the diagonal axis rather than parallel to the axis as would be expected, and that concurrently the other
end of the axis shows a very sudden rise to high values from low values at edge of matrix” (1951:311). This effect is caused by the presence of generally higher agreement coefficients at one end of the time range than at the other. Brainerd interpreted this as evidence of mixing in the deposits, an expectable occurrence in the upper levels of sites that have a long duration. (See further discussion of this point in Chapter III.)

As is indicated in Chapter III, the values in a correlation matrix can be read as elevations above a two-dimensional plane and a topographic map can be drawn from them. Brainerd (1951:Figs. 93, 94) shows what topographic maps of ordered and nonordered matrices should look like. The value of contouring is that one can visualize some characteristics of a matrix more easily than when it is in its numeric form (e.g., Fig. 3-7).

The use of the Brainerd-Robinson method by a number of archeologists has proved instructive. One of the first applications was by Belous (1953), who worked with California burial data and used a mechanical “ordering board” to speed the ordering of the matrix. Belous had too little data from several of the sites, but one should also observe in his example the degree to which arithmetic errors in the handling of the data affect the matrix.

Dixon’s study (1956) of the Snaketown material showed how an attempt at seriation may lead to new interpretations of the primary data. Incidentally, some of his interpretations depended on his analysis of contour plots of the matrix. Flanders (1960) also found that seriation helped him evaluate his Mill Creek ceramic data. Troike’s paper (1957) presents a lucid description of the application of the Brainerd-Robinson technique.

We have not mentioned it specifically before, but plotting a Brainerd-Robinson coefficient matrix is tedious; even with an ordering board the juggling of sites to achieve a good matrix is arduous and time-consuming. These obstacles are, of course, compounded as sites and artifact types are added to the matrix. Even with the aid of a desk calculator, the time required for performing the computations is prohibitive. In addition, manual seriation is susceptible to unconscious human subjectivity.

To cope with this problem and to provide a well-defined seriation procedure, Ascher and Ascher (1963) programmed the method for a digital computer. They reported good results on ethnographic data used by Driver (1956), and on archeological data used by Flanders (1960), and Robinson (1951). Comments on our attempts to use the Aschers’ program appear in Chapter V.

Other recent work on seriation using the Brainerd-Robinson model is reported in a paper by Kuzara, Mead and Dixon (1966) in which a new computer program is described. Like previous studies, this is largely an attempt to devise a computer program suitable for finding order in the data, rather than an examination of the theory and techniques of seriation.
The Meighan Technique

A second approach was set up by Meighan (1959), who had used both the Ford and Brainerd-Robinson techniques with some degree of success and satisfaction, but wanted a method which required a minimum of computation and so could be used in the field. Meighan suggested eliminating all but three major types (or combining similar types to reduce the data to only three categories), then computing the percentage distributions as if these types represented the entire assemblage. He plotted the resulting percentages on triangular coordinate paper and fitted a straight line through the points representing sites to approximate an ordering (e.g., Figs. 5-2, -3, -4).

Actually, a “perfect” series of artifact assemblages may plot as a curve (Figs. 3-3, -4), but this does not matter for practical purposes. With “imperfect” data, the curve closes or the points scatter randomly, in which case a straight line axis cannot be drawn.

After plotting an axis, if one draws lines from the points on the graph to intersect the axis at right angles, the sequence of intersections along the axis will give the chronological order (so long as the trend in the data is related to time). Although the relative order of the sites is thus given, the direction is not—like the Brainerd-Robinson approach, this method does not tell which end is earlier. If we construct ideal cases where the types follow our usual assumptions about change, we find that 3-pole ordering invariably gives the correct chronology. If we test the method with types that have bimodal peaks or “random” behavior with respect to time, we cannot expect to order them chronologically. Since the technique is based on the same assumptions of lenticular change as the Brainerd-Robinson method, it should be used only if the types do conform to that theory. Clearly, one cannot know ahead of time how the types will change, but if there are no trends in the data, the points will show no obvious order and the procedure must either be abandoned or different types must be picked.

The 3-pole analysis was formulated algebraically in a paper by Ascher (1959).

The Dempsey-Baumhoff Technique

An alternate method for finding the best seriation was proposed by Dempsey and Baumhoff (1963). We distinguish two distinct proposals made in their study. First, they criticized the use of percentages of artifacts as in the Brainerd-Robinson method and proposed instead the use of a correlation matrix based entirely on the presence or absence of each type at each of the sites. Second, they proposed a method of “Contextual Analysis” to achieve an ordering. This procedure consists of finding the sites whose scores of agreement diverge the most and using these to make an approximate separation of the sites into an early half and a late half.
Further computations refine the division and result in an ordering based on the mean agreement of each site with the sites in each group. We have chosen to treat Dempsey and Baumhoff's suggestions for the construction of the matrix and the search for an ordering as separate proposals. Therefore, we have used both Brainerd-Robinson and presence-absence matrices with all of the ordering techniques we have examined, including Contextual Analysis.

**Presence-Absence Matrices:** Dempsey and Baumhoff, like Brainerd and Robinson, use a correlation matrix in the process of seriation; theirs, however, is computed on the basis of responses to the question, “Does this type occur either at both of these sites or at neither of them?” The score for each pair of sites is the number of types which give an affirmative answer. It follows that the range of scores of similarity (0-200) which occur in Brainerd-Robinson matrices is larger than the range which is likely to occur in presence-absence matrices, which have a maximum possible score of one less than the number of sites.

If we let

- \( P \) represent the number of sites
- \( T \) represent the number of types
- \( A_{1,j} \) represent the percentage of site \( J \) composed of type \( I \)
- \( X_{i,j} \) represent the correlation between site \( I \) and site \( J \)

then the Presence-Absence correlation matrix may be formally defined:

\[
X_{i,j} = \sum_{k=1}^{T} \begin{cases} 
0 \text{ if } A_{k,i} \times A_{k,j} = 0 \text{ and } A_{k,i} + A_{k,j} \neq 0 \\
1 \text{ if } A_{k,i} \times A_{k,j} \neq 0 \text{ or } A_{k,i} + A_{k,j} = 0 
\end{cases}
\]

for \( I, J = 1, 2, ..., P \).

Dempsey and Baumhoff's objections to the Brainerd-Robinson matrices stem from several sources. First, they argue that types of artifacts whose life span is wholly contained within the duration of the sequence under consideration may confuse a seriation (such artifacts are said to have “Pattern III” distributions). If neither the earliest nor the latest site has examples of some types of artifacts, the seriation may be incorrect because sites in which those types are absent will be regarded as similar even though they are from opposite ends of the chronological scale.

The authors further contend that the Brainerd-Robinson method gives more weight to abundant types than to rare types. As they correctly point out, rare types may be the most sensitive indicators of time. In a presence-absence matrix each type has equal weight; the matrix expresses the pattern of occurrence rather than the pattern of frequency of occurrence.
Contextual Analysis: The second proposal made by Dempsey and Baumhoff relates to the actual process of obtaining a chronological sequence. The first step in ordering is to split the array of sites into two parts. This is done by choosing from the correlation matrix the site which has the greatest range of correlation values (presumably representing one end of the time axis). The sites are ordered according to the correlation values in the column corresponding to this chosen site. The collection is then split into two groups: Group I contains the half of the sites whose correlations with the chosen site are the highest (including the chosen site itself); Group II contains the half of the sites with the lowest correlations. The average agreement of every site with each of the two groups is determined, and the signed difference between the average with Group I and the average with Group II is recorded. The sites are then reordered according to the (signed) differences. If this new order is not consistent with the previous order, the sites are regrouped according to the new order. The process of taking averages and reordering is then repeated until no improvement in the ordering is obtained or a repeating cycle of changes is encountered. The final difference scores then indicate the chronological order in which the sites should occur. In the case that a “final” order does not emerge, Dempsey and Baumhoff average the difference scores of the various orderings in the cycle to obtain the final order.

Returning to their argument about artifacts with Pattern III distributions causing potential error, the authors then attempt to deduce from the order just obtained which items do not vary with time. They set up a table (with the sites in the computed order) showing the behavior of each of the artifacts. If the behavior of an artifact seems inconsistent (based on a statistical criterion of reproducibility), it is tentatively assumed to have a Pattern III distribution and is held aside. A correlation matrix using only the remaining types is then calculated and the previous steps of averaging and reordering are repeated to arrive at a final order.

Dempsey and Baumhoff then make further manipulations of their data to reintroduce Pattern III artifacts into the system. In their example, they found that little improvement resulted from reintroducing these artifacts (Dempsey and Baumhoff 1963: Fig. 2c).

The Permutation Search Technique

After experimenting with other techniques for seriation, we decided to look for an approach that worked directly toward the goals set up by Brainerd and Robinson. The other techniques were not specifically related to the Brainerd-Robinson criteria (e.g., Meighan and Dempsey-Baumhoff), or they simply set up schemes which purported to find the order that fit the Brainerd-Robinson model but did not check with the model in the course of the computation (e.g., Ascher and Ascher). We based our ap-
proach on minimizing the "matrix coefficient," or norm, which Brainerd and Robinson describe, but do not completely specify. We reasoned that we could get the best results by considering the entire matrix at all times and by computing the norm at each step to determine whether we were improving our order. In addition, we set up an economical technique for searching through a series of permutations of the sites to find the ordering which put the correlation matrix into the most nearly ideal shape.

The norm we use is a simple sum of the errors in the matrix—that is, a sum of the differences which are negative as you move away from the diagonal. The norm decreases as the orderings get better, and a norm of zero corresponds to perfect agreement to the ideal. Therefore our approach is to check each of the potential orderings and select the one with the lowest norm.

The only way we know for an automatic procedure to be absolutely certain of having the best ordering inherent in the data (i.e., the lowest norm) is to examine all possible orderings, keeping track of the best one found. Even for high-speed digital computers, the amount of computation required grows prohibitively large as the number of sites increases; this rapid increase in time makes the exhaustive approach to seriation com-

<table>
<thead>
<tr>
<th>Number of Sites</th>
<th>Evaluations for All Orderings</th>
<th>Evaluations for Our Implementation of Permutation Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>n!</td>
<td>(\frac{n(n-1)}{2} + n \times n)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>120</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>720</td>
<td>51</td>
</tr>
<tr>
<td>7</td>
<td>5,040</td>
<td>70</td>
</tr>
<tr>
<td>8</td>
<td>40,320</td>
<td>92</td>
</tr>
<tr>
<td>9</td>
<td>362,880</td>
<td>117</td>
</tr>
<tr>
<td>10</td>
<td>3.63 \times 10^6</td>
<td>145</td>
</tr>
<tr>
<td>11</td>
<td>3.99 \times 10^7</td>
<td>176</td>
</tr>
<tr>
<td>12</td>
<td>4.79 \times 10^8</td>
<td>210</td>
</tr>
<tr>
<td>13</td>
<td>6.23 \times 10^9</td>
<td>247</td>
</tr>
<tr>
<td>14</td>
<td>8.72 \times 10^{10}</td>
<td>287</td>
</tr>
<tr>
<td>15</td>
<td>1.31 \times 10^{12}</td>
<td>330</td>
</tr>
<tr>
<td>16</td>
<td>2.09 \times 10^{13}</td>
<td>376</td>
</tr>
<tr>
<td>17</td>
<td>3.56 \times 10^{14}</td>
<td>425</td>
</tr>
<tr>
<td>18</td>
<td>6.40 \times 10^{15}</td>
<td>477</td>
</tr>
<tr>
<td>19</td>
<td>1.22 \times 10^{17}</td>
<td>532</td>
</tr>
<tr>
<td>20</td>
<td>2.43 \times 10^{18}</td>
<td>590</td>
</tr>
<tr>
<td>21</td>
<td>5.11 \times 10^{19}</td>
<td>651</td>
</tr>
<tr>
<td>22</td>
<td>1.12 \times 10^{21}</td>
<td>715</td>
</tr>
<tr>
<td>23</td>
<td>2.59 \times 10^{22}</td>
<td>782</td>
</tr>
<tr>
<td>24</td>
<td>6.20 \times 10^{23}</td>
<td>852</td>
</tr>
<tr>
<td>25</td>
<td>1.55 \times 10^{23}</td>
<td>925</td>
</tr>
</tbody>
</table>
Computer Analysis of Chronological Seriation

...out of the question for data sets with more than seven or eight sites. Table 2-1 shows the number of orderings which must be evaluated to consider exhaustively various numbers of sites. To put the tabulation into perspective, assume that a computer can run entirely without failure and can evaluate ten orderings per second. Then the evaluation of all possible orders for twelve sites takes over a year; for fifteen sites it would take 50 centuries.

In view of this explosive relationship between the number of sites to be seriated and the number of years it takes to examine all of the permutations, we must restrict ourselves to examining a selection of the possible orders in which the sites may be taken. The Permutation Search approach is our answer to this time problem. The philosophy of the system calls for improving a given ordering by examining a selected set of related orderings which presumably contains the ones most likely to be improvements on the original. The choice of a subset of the set of all possible orderings brings the number of orderings to be examined down into the range of feasibility; this reduction of the search space is justifiable if the selected orderings include the most likely candidates for improvements on the original ordering.

The particular choice of the set of related orderings to be examined can be made by the user of the system; thus it can be varied experimentally or selected to adjust to particular types of data. We believe that the series of manipulations which we have chosen for our application of Permutation Search is a reasonably good one for archeological data; it concentrates on improvements through pairwise interchanges of sites and through trying to fit some site at all possible positions in the sequence formed by the remainder of the sites.

Details of the Search Technique: The first step in a Permutation Search is to specify the set of orderings which will be examined. For a programmed application, it may be most convenient to choose a simple pattern of permutations and provide a method for generating it. A tabulation of the entire sequence of orderings to try, although it will tend to be long and probably useful only for a specific number of sites, will also suffice. To actually apply the technique to a group of sites, the form of correlation matrix for the sites to be ordered must then be supplied. The execution of Permutation Search consists of evaluating the norm in turn for each of the orderings applied to the given data, keeping track of the best order found. When the entire set of orderings has been examined (that is, when the pattern has been carried to completion or the tabulation has been exhausted), one cycle of Permutation Search has been completed.

The seriation which comes out of this cycle must be at least as good as the one that went in, because an ordering is kept only if it is an improve-
ment on all of its predecessors. The order which comes out of the searching cycle is not, however, necessarily the best which can be found by Permutation Search, even by reexamining the same series of permutations. Unless there were no improvements made in the last cycle, there is a good chance that further improvements can be made by rerunning the same sequence of trial orders with the best order found in the last cycle as a starting base from which the pattern of permutations can operate. In practice we have found that it usually takes two or three cycles before no further improvements can be made.

In the course of a cycle, when a trial order is found to be an improvement over all orders already examined, it may be saved in one of two ways: first, it may replace the current basic order for generating permutations (that is, the original order for the current cycle); second, it may be filed away and not referenced again until the end of the cycle. The first scheme, for retaining improved orderings, is called immediate improvement; with it the permutation pattern operates on the best order found so far. The second scheme is called delayed improvement; with it the permutation pattern always operates on the initial order. The latter will probably be used when a tabulation of orders to try is provided.

We have chosen to use two search patterns from among many possible patterns; others can be set up and added to the program with a minimum of effort. For most archeological problems—where we have no previous conceptions of order—it seems advisable to compare the effects of moving single sites within a matrix. If we have prior knowledge that some sites should be paired or otherwise grouped, we can, of course, hold these in the proper positions and allow our series to operate on the remaining sites.

The first of the two sequences of permutations that we use in Permutation Search can be called pairwise interchange. In this series, each trial ordering is derived from its predecessor by interchanging the sites which currently occupy two positions in the ordering. The first trial ordering comes from the starting order by interchanging the sites which occupy the first and second row positions and the first and second column positions; the second trial order comes from the first by interchanging the first and third positions; and so forth until all possible pairwise interchanges have been made. If there are $P$ sites under consideration, this involves $P \times (P-1) / 2$ trials.

The example of the first part of Figure 2-1 shows a complete sequence of interchanges. Programmers will recognize this test sequence as an elementary pattern used for sorting a sequence of numbers into ascending (or descending) order. The difference here is that the method of sorting simple numbers gives the proper result on the first try, while the matrix-evaluation version requires repeated executions until no more improvements can be made.
The second pattern we use for generating trial orderings can be called *successive rotation*. In this series, the emphasis is not on obtaining improvement by interchanging sites, but on finding the best fit for each site relative to the current order for the rest of the sites. If there are \( P \) sites under consideration, the series involves \( P \times P \) trials. The first \( P \) trial orderings are obtained by holding all but the first of the sites in the starting order and testing the first in all \( P \) of the positions in which it can be placed: in front, in all \((P-2)\) slots between sites, and at the end. The second \( P \) trials are generated by doing the same thing with the second site held out to try in all positions, and so forth.

The example of the second part of Figure 2-1 shows a complete sequence of rotations for eight sites. Kuzara, Mead, and Dixon (1966) developed a program which relied entirely upon this series for finding their seriations. Once again, there is no assurance that this will find the best seriation on the first try—it will almost always require two or more iterations.

The relationships among cycle, search patterns, and improvement alternatives is probably best explained with an illustration. Figure 2-1 shows a step-by-step trace for one cycle of a Permutation Search seriation of the example given by Robinson (1951). It is run with the particular form of Permutation Search we have used in most of our work: each cycle consists of one pass through a pattern of pairwise interchanges followed by one pass through a pattern which tries each site in all possible positions with respect to the rest of the sites (successive rotations). The pairwise interchange portion of the cycle is carried out under the immediate improvement alternative for retaining improvement; the successive rotation pattern is carried out under delayed improvement.

In addition to the computer printout which traces the progress of the search (obtained by request, *not* with every seriation run) we show the permutation of the original order which was examined at each step. The numbers in the list of permutations (1 through 8) refer to the sites in their original order (1 represents IIIC, 2 represents IIIB, ..., 8 represents IIA). The order of these numbers in each line denotes the permutation which was checked at that point. Crucial lines in the Permutation Search are keyed to notes in the table.

The lines of the form

```
INTERCHANGE-WITH--; NORM IS NOW--
```

trace the pairwise interchange pattern. The operations should be clear; every position in the sequence is interchanged with every other one at some point in the pattern. Again, it is important to realize that the description "interchange site I with site J" means to examine the order which results from interchanging the sites in the 1st and 1st positions, regardless of their original numbers.
Figure 2-1. Example of permutation search.

Notes:
A Begin cycle of pairwise interchange using immediate improvement.
B Begin cycle of successive rotation using delayed improvement.
C This is the original order for this cycle.
D This trial recognized as an improvement and accepted as such.
E New best order that was just found is used for further trials.
F New best order was held back—old order is used for further trials.
G Improvement over original, but not best found so far.

<table>
<thead>
<tr>
<th>INTERCHANGE</th>
<th>1 WITH 2</th>
<th>NORM IS NOW 1350.2</th>
<th>A</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERCHANGE</td>
<td>1 WITH 3</td>
<td>NORM IS NOW 686.2</td>
<td>D</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>1 WITH 4</td>
<td>NORM IS NOW 1359.0</td>
<td>E</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>1 WITH 5</td>
<td>NORM IS NOW 1574.8</td>
<td>F</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>1 WITH 6</td>
<td>NORM IS NOW 1820.6</td>
<td></td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>1 WITH 7</td>
<td>NORM IS NOW 1377.2</td>
<td>G</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>1 WITH 8</td>
<td>NORM IS NOW 757.4</td>
<td></td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>2 WITH 3</td>
<td>NORM IS NOW 1202.6</td>
<td></td>
<td>9</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>2 WITH 4</td>
<td>NORM IS NOW 1113.6</td>
<td>E</td>
<td>10</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>2 WITH 5</td>
<td>NORM IS NOW 911.2</td>
<td>F</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>2 WITH 6</td>
<td>NORM IS NOW 1411.6</td>
<td></td>
<td>12</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>2 WITH 7</td>
<td>NORM IS NOW 1062.6</td>
<td>G</td>
<td>13</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>2 WITH 8</td>
<td>NORM IS NOW 787.8</td>
<td></td>
<td>14</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>3 WITH 4</td>
<td>NORM IS NOW 571.8</td>
<td></td>
<td>15</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>3 WITH 5</td>
<td>NORM IS NOW 177.4</td>
<td>G</td>
<td>16</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>3 WITH 6</td>
<td>NORM IS NOW 991.4</td>
<td></td>
<td>17</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>3 WITH 7</td>
<td>NORM IS NOW 617.4</td>
<td>G</td>
<td>18</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>3 WITH 8</td>
<td>NORM IS NOW 437.6</td>
<td></td>
<td>19</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>4 WITH 5</td>
<td>NORM IS NOW 597.2</td>
<td></td>
<td>20</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>4 WITH 6</td>
<td>NORM IS NOW 609.0</td>
<td></td>
<td>21</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>4 WITH 7</td>
<td>NORM IS NOW 307.6</td>
<td></td>
<td>22</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>4 WITH 8</td>
<td>NORM IS NOW 739.8</td>
<td></td>
<td>23</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>5 WITH 6</td>
<td>NORM IS NOW 186.2</td>
<td></td>
<td>24</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>5 WITH 7</td>
<td>NORM IS NOW 184.4</td>
<td></td>
<td>25</td>
<td>19</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>5 WITH 8</td>
<td>NORM IS NOW 628.6</td>
<td></td>
<td>26</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>6 WITH 7</td>
<td>NORM IS NOW 177.4</td>
<td></td>
<td>27</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>6 WITH 8</td>
<td>NORM IS NOW 925.2</td>
<td></td>
<td>28</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>INTERCHANGE</td>
<td>7 WITH 8</td>
<td>NORM IS NOW 918.2</td>
<td>G</td>
<td>29</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>PUT 1 IN POSITION</td>
<td>1</td>
<td>NORM IS NOW 177.4</td>
<td>B</td>
<td>30</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>PUT 1 IN POSITION</td>
<td>2</td>
<td>NORM IS NOW 325.0</td>
<td>G</td>
<td>31</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>PUT 1 IN POSITION</td>
<td>3</td>
<td>NORM IS NOW 866.8</td>
<td></td>
<td>32</td>
<td>26</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>PUT 1 IN POSITION</td>
<td>4</td>
<td>NORM IS NOW 841.4</td>
<td></td>
<td>33</td>
<td>27</td>
<td>26</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>PUT 1 IN POSITION</td>
<td>5</td>
<td>NORM IS NOW 1138.8</td>
<td></td>
<td>34</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>PUT 1 IN POSITION</td>
<td>6</td>
<td>NORM IS NOW 1139.6</td>
<td></td>
<td>35</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>25</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>PUT 1 IN POSITION</td>
<td>7</td>
<td>NORM IS NOW 253.2</td>
<td></td>
<td>36</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>PUT 2 IN POSITION</td>
<td>1</td>
<td>NORM IS NOW 279.0</td>
<td></td>
<td>37</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>PUT 2 IN POSITION</td>
<td>2</td>
<td>NORM IS NOW 325.0</td>
<td></td>
<td>38</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>PUT 2 IN POSITION</td>
<td>3</td>
<td>NORM IS NOW 719.2</td>
<td></td>
<td>39</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>PUT 2 IN POSITION</td>
<td>4</td>
<td>NORM IS NOW 693.8</td>
<td></td>
<td>40</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>PUT 2 IN POSITION</td>
<td>5</td>
<td>NORM IS NOW 991.2</td>
<td></td>
<td>41</td>
<td>35</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>PUT 2 IN POSITION</td>
<td>6</td>
<td>NORM IS NOW 992.0</td>
<td></td>
<td>42</td>
<td>36</td>
<td>35</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>PUT 2 IN POSITION</td>
<td>7</td>
<td>NORM IS NOW 248.6</td>
<td></td>
<td>43</td>
<td>37</td>
<td>36</td>
<td>35</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

Table 1

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
</tr>
</tbody>
</table>

Order: 1, 2, 3, 4, 5, 6, 7, 8
## COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION

### COMPUTER TRACE CONTINUED

<table>
<thead>
<tr>
<th>PUT</th>
<th>IV POSITION</th>
<th>NORM IS NOW</th>
<th>ORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT 2</td>
<td>IV POSITION</td>
<td>8</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>1</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>2</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>3</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>4</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>5</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>6</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>7</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 3</td>
<td>LV POSITION</td>
<td>8</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 4</td>
<td>LV POSITION</td>
<td>1</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 4</td>
<td>LV POSITION</td>
<td>2</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 4</td>
<td>LV POSITION</td>
<td>3</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 4</td>
<td>LV POSITION</td>
<td>4</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 4</td>
<td>LV POSITION</td>
<td>5</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 4</td>
<td>LV POSITION</td>
<td>6</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>1</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>2</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>3</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>4</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>5</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>6</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>7</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 5</td>
<td>LV POSITION</td>
<td>8</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>1</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>2</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>3</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>4</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>5</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>6</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>7</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 6</td>
<td>LV POSITION</td>
<td>8</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>1</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>2</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>3</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>4</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>5</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>6</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>7</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 7</td>
<td>LV POSITION</td>
<td>8</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>1</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>2</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>3</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>4</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>5</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>6</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>7</td>
<td>NORM IS NOW</td>
</tr>
<tr>
<td>PUT 8</td>
<td>LV POSITION</td>
<td>8</td>
<td>NORM IS NOW</td>
</tr>
</tbody>
</table>

**ON LOOP 1 VALUE IS 6.0 FOR ORDER AND COLUMN ERRORS**

<table>
<thead>
<tr>
<th>11A</th>
<th>11B</th>
<th>11C</th>
<th>1B</th>
<th>11B</th>
<th>11C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.2</td>
</tr>
</tbody>
</table>

| 1.8 | 0.0 | 0.0 |

8 3 2 5 1 4 7 6
The operation of the immediate improvement scheme for using improved orders may be seen at the four points during this pass (see Figure 2-1, Note D) where a trial order is found to be better than any order previously examined. In each case the next interchanges were made on the basis of the newly improved order. For example, if interchanging 1 with 3 had not produced an improvement, the order tested at the next step (where 1 and 4 were interchanged) would have been

\[4 \ 2 \ 3 \ 1 \ 5 \ 6 \ 7 \ 8\]

instead of \[4 \ 2 \ 1 \ 3 \ 5 \ 6 \ 7 \ 8\].

The lines of the form

\text{PUT — IN POSITION —; NORM IS NOW —}

trace the successive rotation pattern. Again, the operations which produce the series of permutations should be clear. The basic sequence for this set of permutations is

\text{IIIA IIIB IA IIIC IB IIB IIC IIA}

represented by \[3 \ 2 \ 5 \ 1 \ 4 \ 7 \ 6 \ 8\], since that was the best order found by the pairwise interchange section of the search. Each of the sites in this basic sequence is tried at all eight of the positions it can assume.

The operation of the delayed improvement scheme for saving improved orders may be seen at the one point during the pass where a trial order is found to be better than any previously examined. The order with the eighth (starting) position moved to the first position is found to be an improvement and the permutation \[8 \ 3 \ 2 \ 5 \ 1 \ 4 \ 7 \ 6\] is filed away for future reference. Had there been no improvement at that point the next order inspected would have been \[3 \ 8 \ 2 \ 5 \ 1 \ 4 \ 7 \ 6\], just as it was when the improvement was found.

This example shows only the first cycle of the Permutation Search. In this cycle, the norm of the best available seriation was improved from 1178.0 to 6.0. The second cycle produced no further improvement, and is not reproduced here. Since the second cycle left the same order as it had started with, the Permutation Search terminated with a final order of

\text{IIA IIIA IIIB IA IIIC IB IIB IIC}

and a norm of 6.0. This is the order given by Robinson and we, too, consider it to be the correct one.

Another interesting aspect of this seriation is that an improved norm of 177.4 was accepted when positions 6 and 7 were interchanged. According to the printout, this was an improvement over a norm of 177.4, obtained when positions 3 and 5 were interchanged. The explanation is quite simple; the second value was actually an improvement over the first, but the two differed by less than 0.1 and the values were printed only to tenths. Exami-
nation of the later stages of the search shows that the decision to interchange positions 6 and 7 was a wise one. Although there was only a small difference at the time the slight improvement was accepted, the differences in the latter stages of ordering were substantial. This level of precision was made possible because we keep as much accuracy as we can in the percentage tables and correlation matrices. Had we restricted our program to dealing with integers, as others have done and advocated, we would quite possibly have missed this path to the proper seriation.

**Input Order:** In most cases, sites must be put in a preliminary order (say, a guess at the chronological order), but if the set of orderings is to be given as a list rather than a generation pattern, or if the system computes a preliminary order with some other method (see below), the input order can be made irrelevant.

The search patterns we use in Permutation Search share an undesirable feature with almost all search patterns in that the final order is inclined to be dependent on the order with which the search started. If a thoroughly disordered starting sequence is provided, the search may get stuck in a rut from which it will never recover. The most common failure of this type is described below.

One way to minimize the potential deleterious effect of the input order is to start with an order which is in roughly the right sequence. There are several ways to accomplish this preliminary ordering: (1) if the data span a long time, it may be possible to hand sort the sites into sets corresponding approximately to time periods and submit the data ordered by the time periods; (2) independent knowledge about each of the sites may be used; (3) several random input orders may be used (see Kuzara, Mead and Dixon 1966); or (4) some other seriation technique which is neither order-dependent nor good enough to give more than approximate orderings may be used to find a starting order.

We have found that a rough reordering obtained by applying Dempsey and Baumhoff's technique or Meighan's technique to three of the most prevalent types will frequently lead us to a better final ordering or help us find it with less time spent in the Permutation Search proper.

Although we feel that the main emphasis of a search technique should be on the effects of moving single sites, the Permutation Search approach may occasionally allow a situation to develop in which no improvement can be made by moving single sites and, indeed, all changes brought about by single movements are changes for the worse.

In this situation, it is frequently the case that a few large groups of sites are internally well ordered, but the groups of sites are misplaced or inverted with respect to one another. Such situations are easy to detect by examining a contour plot of the correlation matrix for the ordering; the sharp breaks
associated with the point or points at which the major displacement(s) occurs show up as sharp discontinuities in the contour (see Chapter III). The breaks are so striking that it is often sufficient to examine a printout of the correlation matrix on which the locations, but not the magnitudes, of the errors have been marked. Figure 2-2 shows a group of sites in the proper order; Figure 2-3 shows the same sites in an incorrect order. These data are from Robinson’s paper (1951).

The displacement of sets of sites in this way may be caused by major breaks in the data (e.g., two sets of data which are strongly related internally but highly unrelated between groups) or by an extremely poor starting order for the Permutation Search. We do not feel that this is a severe drawback. In the first place, it is easy to find a reversal with even perfunctory examination of the results, and secondly, we feel that one of the main strengths of Permutation Search is its use as a refining process. If other preordering techniques were not available, we could devise other search patterns for Permutation Search which would be good for finding rough orderings.

The proper action to take in case of major reversals of the type described above is to set up a new starting order based on intuitively sensible rear-
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION

rangement of the major groups, without changing the order within the groups, and then perform another Permutation Search based on the new order. In the example of Figures 2-2 and 2-3, a reasonable trial reorder would be

IIA IIIA IIIB IA IIIC IB IIB IIC

which happens in this case to be the best known order.

Figure 2-3. Contour plot of correlation matrix in incorrect order (Robinson's data). Order of sites in matrix is indicated along left margin of the plot.

Constraints: This sort of situation illustrates an important point about seriation techniques in general and Permutation Search in particular: Seriation is still as much an art as a science, and automatic procedures should be regarded as conversational aids in the search for an ordering, not as infallible, all-seeing oracles.

We do not believe that a mechanical procedure can be devised to give the best of all possible orderings in all cases for data of the quality which can now be obtained. There is still a need for informed human intervention in the ordering process; the best chronologies can be most consistently obtained by allowing human judgment to play a part in the seriation process.

We differ from Kuzara, Mead and Dixon (1966) in this respect. They attempt to find the best order inherent in the data, assuming the theory of
systematic change in artifacts and that the data are suitable. The best order therefore becomes the chronological order. This approach satisfies the purist but not the pragmatist. It was evident after we seriated many sets of test cases that the best order was not always the correct order. As a result we feel strongly that when archeologists have any firm indication of chronology (from whatever independent source) they ought to use that knowledge as a first step and only allow automatic techniques to operate within the predetermined constraints. This specification of chronology seems as natural a part of the set of data as the listing of counts of types in each site.

There are two classes of situations in which seriation techniques are likely to be useful and for which the constraints will normally be different. The first is the situation generally treated in the literature: the user has a number of sites for which he knows almost nothing (or equally little for each site) about the correct chronology. In this case, it is probably not desirable to impose constraints on the ordering.

The second case involves having a preestablished chronology for a substantial number of sites in the cultural region in question and a number of new ones to fit into the existing order. (The preestablished order must of course be a good ordering according to the criteria of the program.) In this case, we believe it is most meaningful to see where the new sites fit into the given sequence, perhaps with additional unconstrained permutation searches to check on the stability of the result. As an additional comment, we note that considerations of computation time suggest constraining the order of the "known" sites, for this reduces the number of orderings which must be considered in the search.

Although a feature for specifying constraints on the orderings is included in the specification of Permutation Search, it has not yet been added to either of the realizations of PHOENIX. We feel, however, that it should be added at some future time: the system is not complete without it.

The Type-Percentage Technique

Most of our studies sought an ordering of a correlation matrix, but it is possible to work more directly with the data, still using the same assumptions about change. We might try this by measuring the performance of each type against our theoretical expectations of lenticular change (instead of the pattern of a whole matrix). This procedure keeps the computation closer to the actual data than the correlation matrix does and although we have no particular misgivings about correlation matrices, a more direct procedure is aesthetically more satisfying.

In detail the procedure, called the Type-Percentage technique, is to minimize the following norm over the percentage matrix (percentages of each type in each site): Let A be the percentage matrix; each $A_{IJ}$ is the percentage of type I at site J. The norm is defined, again as a deviation from
an ideal, by

\[
\sum_{i=1}^{\text{types}} \sum_{j=1}^{\text{sites}-1} |A_{i,j} - A_{i,j+1}| \Delta
\]

where \( \Delta = 0 \) if the percentages go the right way (ascending toward the peak) and \( \Delta = 1 \) if the percentages go the wrong way.

Thus a norm penalizes a type at every place where its pattern of percentages deviates from the desired; the size of the penalty is equal to the size of the error in the method we use here. The problem which arises, of course, is defining the peak from which everything else is supposed to slope—since the sense of the test ('>' or '<') is changed as the peak is passed. This can be done by requiring the programmer to specify for each type which site contains the peak; another possibility is to search for the maximum percentage over all the sites; a third approach is to pick the two or three points of highest frequency for ambiguous types and try all of them.

The technique for finding the best order is a simple variant on the Permutation Search procedures described above. Instead of computing norms from a Brainerd-Robinson or Presence-Absence matrix, the Type-Percentage technique computes a norm directly from a percentage matrix. Aside from this change, the Type-Percentage and Permutation Search techniques are identical.

The key to the Type-Percentage method is that, theoretically, each type changes in a lenticular pattern. Therefore, a lensing pattern cannot be detected unless a type occurs in at least three sites, but if a type occurs in two sites, they should be placed adjacent to one another. By contrast, in both types of matrix, rare types can be helpful. The Type-Percentage method has an advantage over correlation matrices if one type is very large numerically. In Data Set II our problem with one type of flint dominating the orders would not have been so severe.

A practical limitation of the Type-Percentage method is that adding types adds more or less proportionately to the running time of the program. For this reason it is desirable to keep the number of types fairly small. With this in mind, we note that of the 33 types considered in Data Set II, only 16 occurred in three or more components.

In addition to discarding unsuitable types, we can also lump similar ones; however, this may obscure just the fine distinctions that the typology originally brought out and it should only be done with great caution. The same can be said for omitting types that are relatively rare. It may happen that in a long series the most sensitive types are those that cluster in only three or four components somewhere in the time scale; these should not necessarily be omitted in favor of types that are found in twice as many components. In cases like these, the judgment of the archeologist is the
only safeguard against possible error.

The Type-Percentage method has been programmed but we have not yet made an extensive study of its characteristics.
CHAPTER III
EVALUATING RESULTS

In this section we consider four separate topics that bear on the interpretation and evaluation of results. We consider matrix norms, which may be regarded as a tool for evaluating possible chronological orders for sites; error coefficients, which may be considered tools for evaluating an ordering or seriation technique when we know by other means what the order is; characteristics of data, where we discuss the quality of data and their suitability for seriation; and display of results, where we show how two forms of graphic representation of matrices help in evaluating the performance of the ordering and the nature of the data.

Matrix Norms

Norms are used as simple numerical indices to rank the possible orders for a matrix. Norms are easy to compute and they are objective but there is some discussion among archeologists over the relative merits of the various norms which have been proposed. However, all of the norms we have considered have similar characteristics. In all cases, the norm is zero for orderings which agree perfectly with the ideals, is small for good approximations, and grows larger as the orderings get worse or as noise in the data increases. In other words, higher norms indicate a deviation from the ideal somewhere in the system.

The value of a norm is computed from the “errors” in a matrix. An error occurs when the difference between some matrix entry and the entry immediately adjacent to it as you leave the diagonal along a row or column is negative instead of positive. Since the matrices are symmetric, it is sufficient to consider only differences within rows (or within columns). This ensures that an error is counted once instead of twice. We call this measurement a “norm” even though this does not conform to precise mathematical usage of the term.

Note that an ordering need not have been derived from a seriation technique employing a matrix in order to have its correlation matrix evaluated with a matrix-based norm. This means that we can find norms for 3-pole or other orderings so long as we have the data from which to compute a correlation matrix.

Proposals for Norms: Robinson considered two matrix norms based on
the errors among the matrix entries. The first was a *simple count of errors in the matrix* (Robinson 1951:298); the second took into account the size of the errors, and was defined as the *sum of the squares of the errors* and divided by the *sum of squares of all differences* (Robinson 1951:300). A third possible norm, which gives relatively less weight to very large errors, is a *simple sum of errors*. Finally, we might consider a norm which is the *sum of squares of errors*.

Lehmer (1951) criticized Robinson’s choice of agreement coefficient on the grounds that it makes no allowance for differences in the sizes of the collections. He proposed in its stead a *mean standard error*, which includes sample sizes in the formula for the coefficient. Robinson and Brainerd (1952) defended their choice, offering an example to justify their use of the sum of the percentage differences. According to Robinson (1951:300) in his earlier paper,

... it may be necessary to use a more sensitive measure of the agreement between the actual and the desired pattern of agreement coefficients than is provided by the number of negatively signed differences. There appears to be no good theoretical ground upon which such a measure can be based, but we have used one which serves very well in practice. It seems obvious that the size of a difference ought to count in assessing the agreement between a given matrix and the desired one as well as the sign of the difference. A negative difference of one or two units is obviously not as damaging as a negative difference of a hundred units. For very precise work we therefore use such an index of disagreement between the given and the desired matrix. We square all the negative differences, and then express the *sum of these squares* as a fraction of the sum of the squares of all differences, positive and negative together. The reason for using the squares of the differences is that the squares give proportionately more importance to larger differences.

*Our Choice of Norm:* In view of the various opinions on the merits of particular norms, we shall explain our choice—*a simple sum of the errors*—more fully. There are two important considerations: first, whether the norm expresses what it is supposed to, and second, whether it can be readily computed.

The choice of a norm depends on the emphasis you wish to give to different types of errors that may show up in a matrix. A chronological series deduced from a matrix that contains only perfect data in perfect order will yield a norm of 0. If we recognize that there are few perfect sets of data, we must devise and make use of a norm that works toward an ideal pattern for the matrix, penalizing orderings only on archaeologically valid grounds.

Errors in a matrix stem from two sources: from the typological classification of the artifacts, and from the array of artifacts that are included in the typology. Since the correlation coefficients on which the norms are based compare the percentages of types found in each component, the quality of the typology (as explained in Chapter I) bears directly on the correlation coefficients. Consequently, in choosing a norm it seems inadvisable to us to use one that emphasizes minor errors in typology.

The second source of error is the data set itself. Our matrix model is
based on a closed system in which factors other than time are controlled. In this model, contemporary sites have identical assemblages of artifacts. In fact, however, we know that most social systems are diverse enough, even if not completely open, to permit variation among sites fairly close in time and space. Such factors as local craft specialization, variation in technical skill among artisans, and trade or borrowing may lead to differences in assemblages of artifacts in different contemporary sites.

Added to these sources of error is the possibility that a given sample from a site is not adequate. Recognizing these factors, however, we know that many differences among sites are likely to be related to time. Our strategy, therefore, is to use a norm which will seek the proper pattern without becoming too confused by aberrant data.

In our opinion, the norm that computes a simple sum of errors conforms most closely to our appreciation of the cultural problems. We do not argue that errors are unimportant; on the contrary, they lead us to close inspection of our data and help us evaluate our orderings. But if large errors are rated too highly they create static in a matrix which makes the pattern hard to read and tends to throw the better components out of sequence.

A norm which has this undesirable effect is one used by Brainerd (1951) and Robinson (1951) and also by Kuzara, Mead and Dixon (1966). This norm is computed by squaring the sums of negative differences. By this method, an error of 5 becomes 25 times as bad as an error of 1 does with our norm; conversely, the severity of an error of 1 is thus rated as only 4% of the severity of an error of 5. In short, this norm emphasizes just the kinds of errors we wish to de-emphasize.

Comparing Different Matrices: The numerical values of norms depend on four variables: (1) the data base, (2) the method used for computing the correlation coefficients, (3) the type of norm, and (4) the deduced order of the sites. For this reason we cannot necessarily base our judgment of the relative validity of two orderings simply on the values of their norms: we can only do so when the first three variables are held constant. The value of the norm therefore depends solely on the order of the sites. In the tables in Chapter IV we give the norms for runs made with different algorithms on different forms of matrices, using different data sets. Even though the values of the norms for these orderings may vary greatly, we cannot use this information to judge the relative quality of the orderings.

The data base is subject to more variation than the other factors. The number of sites, the number of types, the sizes of samples, and the choice of types used in the particular data sets all affect the magnitude of the correlation coefficients. Moreover, the range of (apparent) time covered by the data will affect the values of the entries—short time spans show consistently higher correlations with relatively more, but generally smaller, er-
rors (Table 3-1 and Fig. 3-1). As we have already mentioned, different computations of the correlation coefficients emphasize completely different aspects of the data and thus affect the magnitude of the norm. Finally, the form of the norm itself can have a great effect on the magnitude of its values. A simple tally of errors gives norms of lower magnitude than a norm which takes into account the magnitude of errors, and it in turn gives lower values than a norm which squares the sums of the errors.

TABLE 3-1

<table>
<thead>
<tr>
<th>Range of Matrix Values</th>
<th>Data Set I BR</th>
<th>±</th>
<th>Data Set II BR</th>
<th>±</th>
<th>Data Set III BR</th>
<th>±</th>
<th>Data Set IV BR</th>
<th>±</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-9</td>
<td>21.3</td>
<td></td>
<td>52.6</td>
<td></td>
<td>45</td>
<td></td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>10-19</td>
<td>4.7</td>
<td>3.6</td>
<td>2.1</td>
<td>47.4</td>
<td>5.5</td>
<td>10.3</td>
<td>12.4</td>
<td></td>
</tr>
<tr>
<td>20-29</td>
<td>22.5</td>
<td></td>
<td>4.2</td>
<td></td>
<td>1.7</td>
<td></td>
<td>11.8</td>
<td></td>
</tr>
<tr>
<td>30-39</td>
<td>32.0</td>
<td>7.6</td>
<td>1.2</td>
<td></td>
<td>12.5</td>
<td></td>
<td>11.1</td>
<td></td>
</tr>
<tr>
<td>40-49</td>
<td>15.4</td>
<td>10.1</td>
<td>10.4</td>
<td></td>
<td>11.0</td>
<td>8.3</td>
<td>23.5</td>
<td></td>
</tr>
<tr>
<td>50-59</td>
<td>4.7</td>
<td>11.8</td>
<td>8.9</td>
<td></td>
<td>2.0</td>
<td>22.1</td>
<td>4.8</td>
<td>23.5</td>
</tr>
<tr>
<td>60-69</td>
<td>8.3</td>
<td>8.9</td>
<td>1.7</td>
<td></td>
<td>4.8</td>
<td>5.5</td>
<td>16.6</td>
<td></td>
</tr>
<tr>
<td>70-79</td>
<td>9.5</td>
<td>5.5</td>
<td>3.4</td>
<td></td>
<td>5.5</td>
<td>5.5</td>
<td>16.6</td>
<td></td>
</tr>
<tr>
<td>80-89</td>
<td>5.9</td>
<td>2.8</td>
<td>3.7</td>
<td></td>
<td>2.1</td>
<td>5.5</td>
<td>5.5</td>
<td></td>
</tr>
<tr>
<td>90-99</td>
<td>5.9</td>
<td>4.2</td>
<td>2.1</td>
<td></td>
<td>1.4</td>
<td>5.9</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>100-109</td>
<td>2.4</td>
<td>5.5</td>
<td>4.1</td>
<td></td>
<td>1.4</td>
<td>5.9</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>110-119</td>
<td>2.4</td>
<td>7</td>
<td>11.6</td>
<td></td>
<td>1.4</td>
<td></td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>120-129</td>
<td>2.4</td>
<td>4.8</td>
<td>11.9</td>
<td></td>
<td>1.4</td>
<td></td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>130-139</td>
<td>1.2</td>
<td>3.5</td>
<td>9.9</td>
<td>11.4</td>
<td>1.4</td>
<td>5.9</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>140-149</td>
<td>2.5</td>
<td>4.8</td>
<td>5.4</td>
<td>8.3</td>
<td>1.4</td>
<td>5.9</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>150-159</td>
<td>3.6</td>
<td>3.5</td>
<td>13.2</td>
<td>11.9</td>
<td>1.4</td>
<td>5.9</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>160-169</td>
<td>1.2</td>
<td>7</td>
<td>8.3</td>
<td>11.9</td>
<td>1.4</td>
<td>5.9</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>170-179</td>
<td>7.7</td>
<td>5.9</td>
<td>4.5</td>
<td>289</td>
<td>169</td>
<td>289</td>
<td>484</td>
<td>289</td>
</tr>
<tr>
<td>180-189</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>190-199</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We have considered some ways to compensate for the noncomparability of norms but none of the proposals has been entirely satisfactory. (Naturally, all these proposals assume that the norm is being computed in the same way.) If we divide the norm by the square of the number of sites, we can compensate somewhat for differences in matrix size, but this does not allow for differences in the data base itself. Robinson's proposal for dividing by the sum of squares of all differences brings the norms to approximately the same value, but again does not compensate for differences in the data base.

In this paper, we have restricted ourselves to ranking only those norms that are computed by the same method, using the same correlation matrix from different orderings of the same data.
Figure 3-1. Histogram of frequency distribution of values in correlation matrices.

The unshaded parts of the histogram (□) represent entries in the Brainerd-Robinson correlation matrices (BR) and the shaded parts of the histogram (■) represent entries in Presence-Absence ± matrices.

**Computation Time**

The choices of a norm and of procedures of automatic seriation are influenced by considerations of computation time. The actual costs in computation time will vary with the machine and with the program, but the figures given here are typical.

Although a digital computer performs arithmetic operations rapidly, the seriations frequently involve tens of thousands of operations, and it is desirable to give some attention to the efficiency of the calculation. The speed of Permutation Search, in particular, is limited by the length of time it takes to evaluate the norm of a trial ordering; the computation of the norm is dominated in turn by the arithmetic operations. An accepted method for comparing running times of such programs is to compare the number of additions it takes to perform an evaluation: a subtraction is rated equal to an addition; a multiplication or a division is rated as equal to three additions. The relative costs in machine time for matrices of various sizes are shown in Table 3-2; these values are plotted in Figure 3-2.

In order to obtain definite values in the tables instead of values expressed in terms of the number of errors in the matrices, we have assumed
### TABLE 3-2
Cost of Computing Various Matrix Norms

<table>
<thead>
<tr>
<th>P = 0.1</th>
<th>N</th>
<th>#err, err</th>
<th>err²</th>
<th>err²/diff²</th>
<th>P = 0.25</th>
<th>N</th>
<th>#err, err</th>
<th>err²</th>
<th>err²/diff²</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>2.20</td>
<td>2.80</td>
<td>13.20</td>
<td>2</td>
<td>2.50</td>
<td>3</td>
<td>4.00</td>
<td>13.50</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6.60</td>
<td>8.40</td>
<td>33.60</td>
<td>3</td>
<td>7.50</td>
<td>12.00</td>
<td>12.00</td>
<td>34.50</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>13.20</td>
<td>16.80</td>
<td>64.20</td>
<td>4</td>
<td>15.00</td>
<td>24.00</td>
<td>24.00</td>
<td>66.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>22.00</td>
<td>28.00</td>
<td>105.00</td>
<td>5</td>
<td>25.00</td>
<td>40.00</td>
<td>40.00</td>
<td>108.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>33.00</td>
<td>42.00</td>
<td>156.00</td>
<td>6</td>
<td>37.50</td>
<td>60.00</td>
<td>60.00</td>
<td>160.50</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>46.20</td>
<td>58.80</td>
<td>217.20</td>
<td>7</td>
<td>52.50</td>
<td>84.00</td>
<td>84.00</td>
<td>223.50</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>61.60</td>
<td>78.40</td>
<td>288.60</td>
<td>8</td>
<td>70.00</td>
<td>112.00</td>
<td>112.00</td>
<td>297.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>79.20</td>
<td>100.80</td>
<td>370.20</td>
<td>9</td>
<td>90.00</td>
<td>144.00</td>
<td>144.00</td>
<td>381.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>99.00</td>
<td>126.00</td>
<td>462.00</td>
<td>10</td>
<td>112.50</td>
<td>180.00</td>
<td>180.00</td>
<td>475.50</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>121.00</td>
<td>154.00</td>
<td>564.00</td>
<td>11</td>
<td>137.50</td>
<td>220.00</td>
<td>220.00</td>
<td>580.50</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>145.20</td>
<td>184.80</td>
<td>676.20</td>
<td>12</td>
<td>165.00</td>
<td>264.00</td>
<td>264.00</td>
<td>696.00</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>171.60</td>
<td>218.40</td>
<td>798.60</td>
<td>13</td>
<td>195.00</td>
<td>312.00</td>
<td>312.00</td>
<td>822.00</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>200.20</td>
<td>254.80</td>
<td>931.20</td>
<td>14</td>
<td>227.50</td>
<td>364.00</td>
<td>364.00</td>
<td>958.50</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>231.00</td>
<td>294.00</td>
<td>1074.00</td>
<td>15</td>
<td>262.50</td>
<td>420.00</td>
<td>420.00</td>
<td>1105.50</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>264.00</td>
<td>336.00</td>
<td>1227.00</td>
<td>16</td>
<td>300.00</td>
<td>480.00</td>
<td>480.00</td>
<td>1263.00</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>299.20</td>
<td>380.80</td>
<td>1390.20</td>
<td>17</td>
<td>340.00</td>
<td>544.00</td>
<td>544.00</td>
<td>1431.00</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>336.60</td>
<td>428.40</td>
<td>1563.60</td>
<td>18</td>
<td>382.50</td>
<td>612.00</td>
<td>612.00</td>
<td>1609.50</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>376.20</td>
<td>478.80</td>
<td>1747.20</td>
<td>19</td>
<td>427.50</td>
<td>684.00</td>
<td>684.00</td>
<td>1798.50</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>418.00</td>
<td>532.00</td>
<td>1941.00</td>
<td>20</td>
<td>475.00</td>
<td>760.00</td>
<td>760.00</td>
<td>1998.00</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>462.00</td>
<td>588.00</td>
<td>2145.00</td>
<td>21</td>
<td>525.00</td>
<td>840.00</td>
<td>840.00</td>
<td>2208.00</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>508.20</td>
<td>646.80</td>
<td>2359.20</td>
<td>22</td>
<td>577.50</td>
<td>924.00</td>
<td>924.00</td>
<td>2428.50</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>556.60</td>
<td>708.40</td>
<td>2583.60</td>
<td>23</td>
<td>632.50</td>
<td>1012.00</td>
<td>1012.00</td>
<td>2659.50</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>607.20</td>
<td>772.80</td>
<td>2818.20</td>
<td>24</td>
<td>690.00</td>
<td>1104.00</td>
<td>1104.00</td>
<td>2901.00</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>660.00</td>
<td>840.00</td>
<td>3063.00</td>
<td>25</td>
<td>750.00</td>
<td>1200.00</td>
<td>1200.00</td>
<td>3153.00</td>
</tr>
</tbody>
</table>

N = Number of sites
P = Probability of some difference being an error

For an N × N matrix, there are D = N(N-1) differences to consider; of these D differences, D × P = N(N-1)P are errors.

<table>
<thead>
<tr>
<th>Type of Norm</th>
<th>Symbol</th>
<th>Computation Cost (Additions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple error count</td>
<td>#err</td>
<td>D + DP</td>
</tr>
<tr>
<td>Sum of error values</td>
<td>err</td>
<td>D + DP</td>
</tr>
<tr>
<td>Sum of squares of errors</td>
<td>err²</td>
<td>D + 3DP + DP</td>
</tr>
<tr>
<td>Sum of squares of errors/sum of squares of differences</td>
<td>err²/diff²</td>
<td>D + 3D + D + DP + 3</td>
</tr>
</tbody>
</table>

average error densities for the matrices of one error every ten points and one error every four points (Table 3-2).

For comparison of the computation time required for the evaluation of the various norms, let

- N represent the number of sites in the data set
- D represent the number of differences which must be considered
  Note: D = N × (N - 1)
- P represent the probability that some difference is an error
- Xᵢ,ⱼ represent the correlation of site I with site J

and define

\[ \Delta (i, j) = \begin{cases} 
0 & \text{if the step from } X_{i,j} \text{ to } X_{i,j+1} \text{ is in the right direction (i.e., ascending toward the diagonal)} \\
1 & \text{if the step from } X_{i,j} \text{ to } X_{i,j+1} \text{ is in the wrong direction (i.e., descending toward the diagonal)} 
\end{cases} \]
then the required computation times are given:

Simple Error Count:

$$\#\text{err} = \sum_{I=1}^{P} \sum_{J=1}^{P-1} \Delta (I, J)$$

Sum of Errors:

$$\sum_{I=1}^{P} \sum_{J=1}^{P-1} |X_{I,J} - X_{I,J+1}| \Delta (I, J)$$

For both of these norms, the computation time required is one subtraction per difference (D) to determine whether there is an error, plus one addition per error (D \times P) to accumulate the sum of errors. Thus the computation time required is $D + D \times P$. 
Sum of Squares of Errors:

\[ \sum \text{err}^2 = \sum_{I=1}^{P} \sum_{J=1}^{P-1} \left| X_{I,J} - X_{I,J+1} \right|^2 \triangle (I,J) \]

The time required for this norm amounts to one subtraction per difference (D) to determine whether there is an error, plus one multiplication per error (3 x D x P) to obtain its square, plus one addition per error (D x P) to accumulate the penalty. Thus the computation time required is D + 3 x D x P + D x P.

Sum of Squares of Errors divided by Sum of Squares of all Differences:

\[ \frac{\sum \text{err}^2}{\sum \text{diff}^2} = \frac{\sum_{I=1}^{P} \sum_{J=1}^{P-1} \left| X_{I,J} - X_{I,J+1} \right|^2 \triangle (I,J)}{\sum_{I=1}^{P} \sum_{J=1}^{P-1} \left| X_{I,J} - X_{I,J+1} \right|^2} \]

To compute this norm requires one subtraction per difference (D) to determine whether there is an error, plus one multiplication per difference (3 x D) to obtain the square of the difference, plus one addition per difference (D) to accumulate the sum of squares of all differences for the denominator, plus one addition per error (D x P) to accumulate the sum of squares of errors for the numerator, plus one division (3) to obtain the final quotient. Thus the computation time required is D + 3 x D + D + D x P + 3, plus a little extra bookkeeping.

Error Coefficient

There are cases when the order of at least part of the data is known and it is desirable to obtain a measure of agreement between the order deduced by various algorithms and the known order. These cases arise when we are evaluating algorithms, when we are evaluating data to test their suitability in seriation, or when we are attempting to fit new data of unknown age into an established ordering.

We considered a number of candidates for an “error coefficient” before settling on the one we use in this paper. Since there are arguments for and against each error coefficient, we review four possibilities. Like the norm, the error coefficient is 0 when the orderings agree perfectly with the known order, and grows larger the more the deduced orderings diverge from the known order.

Let V be the ordering to be evaluated, with each site represented by its position in the known order—if V_J is K, then the Kth site in the real
order is in the Jth position in the trial order. With this in mind, we describe four possible error coefficients.

1. **Sum of Distances Between Adjacent Elements**

\[ \sum_{J=1}^{N-1} \left( |V_J - V_{J+1}| - 1 \right) \]

After examining various error coefficients this form seemed the most desirable because it was most often in agreement with what we judge are the best orders. As a sum of errors between adjacent elements, it is consistent with our choice of a matrix norm. It is not dependent on a good ordering over most of the sequence to measure an error at any given point, and it rewards any correct ordering of subsets that exists, whether or not the subsets are properly placed. This is a particular strength of the error coefficient because if a data set spans a long time, it may contain one or more major discontinuities or reversals. Automatic seriation techniques which are normally effective may find reasonably good orderings for separate portions of the data, but misplace entire subsets or invert their orders when combining them. Such major reversals are fairly easy to recognize in the displays of output, and we do not feel that an ordering should be penalized as heavily for inversion of a phase as for indiscriminant scrambling of sites.

2. **Sum of Displacement of Elements**

\[ \sum_{J=1}^N \left| V_J - J \right| \]

The chief difficulty of error coefficient based on the sum of displacements of individual elements from their correct positions is that it puts a high value on placing a site in a particular slot and penalizes internal consistency of subsets if they are shifted from their correct positions or inverted with respect to the remainder of the sequence. There may also be a minor problem in determining which end of the proposed ordering is supposed to be early. The most charitable solution to this problem is to evaluate any questionable orderings both ways and pick the smaller value.

3. **Number of Transpositions**

This method counts the number of transpositions of adjacent elements necessary to achieve the known order. It suffers from the same difficulties as the measure of displacements, with the added problem that it is difficult to compute.

4. **Changes of Direction**

This method calls for counting the number of times the given ordering
changes direction with respect to the known order. That is, how often does one of these pairs of inequalities hold:

\[ V_{j-1} < V_j \quad \text{but} \quad V_j > V_{j+1} \]
\[ V_{j-1} > V_j \quad \text{but} \quad V_j < V_{j+1} \]

This method has the advantage of rewarding internal consistency much as the sum of distances method does but it does not always penalize inconsistency. Moreover, the resultant values are not as large; hence it is more difficult to discriminate among orderings.

**Characteristics of Data**

For purposes of seriation, types of artifacts are grouped into sets of data which can be rated as good or bad depending on how they seriate. Good data conform to the theoretical model of change and always lead to a correct seriation. Bad, or “ill-conditioned,” data do not lead to good seriations. When we use the terms “good” and “bad” with reference to data, we refer only to their value in seriation and not to their worth for other purposes.

Ill-conditioned data may be characterized by a number of factors: (1) little variability in the time involved, (2) random occurrence in the se-

![Figure 3-3. Percentage distribution for three ideal types.](image)
quence, (3) nonlenticular pattern of change. If the types exhibit these factors, it is not possible to achieve good orderings with the methods we have outlined here.

**Recognizing Good and Bad Data:** In the absence of independent methods of ordering, how can one tell which data sets are suitable? As we point out in Chapter V, some types of artifacts seem inherently better for seriation than others, but we are concerned here with individual data sets themselves.

The simplest test we have found for predicting whether data will seriate well is to construct a Meighan 3-pole graph from three of the types. If the plotted points spread out along a line rather than clump with no obvious order, then there is some order in the data—and indeed the plot may be used to determine an approximate order for the data. If the points show no signs of defining a line, the data may not conform to our lenticular model of change but before abandoning attempts to seriate such a data set, it is prudent to change one of the types and try again.

As Meighan notes (1959:204), the points on the graph need not lie along a straight line in order to conform to our theory of change. Figures 3-3 and 3-4 show a set of ideal types which change in a lenticular fashion, but which yield decidedly curved 3-pole plots. Meighan anticipated the problem of getting a curved axis when he described the procedure for drawing the axis. “In the present state of knowledge there is no way of
knowing the direction or degree of curvature that fits the situation. For this reason it is easiest to draw a straight line through the clustered points in such a way that there is approximately equal number of points on each side of the line and all of the points are as close to the line as possible" (1959:204).

It sometimes happens that a clear break between two portions of the data is indicated. If so, the two sections may be treated separately and fit by two ordering lines. This behavior indicates that for the types used the data contains a discontinuity, but it does not necessarily follow from this that there is a discontinuity in the data set as a whole.

The 3-pole test is not, of course, foolproof. Data may plot along a line without being in order; the existence of a clear line of ordering does, however, indicate that the data are potentially good. (See further discussion of this in Chapter V.)

Generally speaking, matrices in which there occurs a large, rather than a small, range of values are better for seriation. Unfortunately, it is difficult to specify limits. In a Brainerd-Robinson matrix the range of possible values is 0-200; in a Presence-Absence matrix the values can range only up to the number of types and they are constrained to be integers. If in either type of matrix there is a high degree of correlation throughout, the values may cluster in a rather narrow range and make discrimination difficult. We conclude that it is hard to judge the quality of a data set from an unmodified correlation matrix.

A good test with seriation techniques that are sensitive to input order is to take the same data set and run it repeatedly with different input orders. If most of the runs give similar orders, the data are probably well conditioned. The same is true if the data set is run with several different programs: if order is not obvious in the data, the different programs will yield different results. Our tests have shown that even poorly designed programs will handle test cases correctly if the data are quite good.

One can also examine the errors to estimate how closely the data conform to the model the program is attempting to achieve. Here again, as we explained earlier, we cannot give numerical limits for a "permissible" level of errors because the size of the errors depends on the characteristics of the data and the matrix.

It should be clear from the foregoing that we have found no absolutely certain ways of identifying poor data sets simply by examining their characteristics, but we have found diagnostic tests which can suggest the presence of poor data.

Improving Poor Data Sets: If one has some independent way to order some of the data and the set as a whole seems poor for seriation, there may be ways to improve it. Poor data sets can often be improved by eliminating
the troublesome types and/or substituting new ones. In the examples used in this paper we have tried to improve some of our data sets by removing types, by lumping types, and by combining components.

For example, in running the Pa Sangar Data (Data Set III), we did not obtain enough discrimination in our original matrix and the flint did not seriate as well as we wished. We determined that this resulted from the fineness of the stratigraphic cuts which produced a matrix that had a high degree of correlation among all components. We solved this problem by lumping components. When lumping of components is not desirable, as in Data Set II, we found that we could improve our results by eliminating some of the types which were not suited for seriation.

Improving data sets requires some independent knowledge either of the behavior of certain types or of the correct order and, unfortunately, this kind of information is ordinarily not available.

Display of Results

In addition to printing tables of numbers, we can convert the results into graphic forms for additional help in interpretation. Although theoretically there is nothing in the graphic computer printouts that is not displayed in the numerical tables, we find it easier to evaluate our data and results if we can study contour and percentage plots of the matrices. We generate these

![Figure 3-5. Contour plot of a perfect sequence of 20 sites with uniform spacing of intervals. Order of sites in matrix is indicated along left margin of the plot.](image-url)
plots on the computer for reasons of speed (an afternoon's work can be reduced to 30 seconds), and because it is difficult to construct an objective plot by hand.

Contour Plots: In essence, contour plots are graphic representations of the numerical patterns we obtain in the correlation matrices. If we treat the matrix values as units of elevation above a plain, the decrease in value as we leave the diagonal plots is a slope away from a central ridge. In other words, we draw a topographic map and by varying the contour interval we can achieve different degrees of detail. This map can be interpreted much more quickly than a matrix of numbers when we wish to get a feel for the performance of the data.

A little practice is required to interpret the contour plots and it is helpful to begin by considering plots of perfect data.

1. Perfect sequences with uniform intervals of differences will plot as a series of lines parallel to the diagonal (Fig. 3-5).

2. A continuous series with nonuniform intervals will plot as a series of lines varying in distance from one another and veering closer or farther from the diagonal as the differences become greater or smaller (Figs. 3-6

Figure 3-6. Contour plot of a matrix computed from the data used in Figure 3-5 (with sites numbered 5, 9, 10, 14, 15, and 16 removed, leaving breaks of 2, 3, 4 intersite units). When the remaining sites are evenly spaced across a correlation matrix, the omitted sites make their absence known by breaks in the contour lines. Order of sites in matrix is indicated along left margin of the plot.
and 4-1). The presence of sharp breaks does not necessarily mean that the data or the seriation are not good; it may simply mean that transitional sites were not included in the data set.

3. Clusters of similar values in the matrix will plot as plateaus or level areas (Fig. 3-7).

![Figure 3-7. The best order found for Data Set I. Order of phases in matrix is indicated along left margin of the plot.](image)

NOTE: (1) Three plateaus along the diagonal in lower right; these correspond to three pottery phases at Tepe Sabz (see description with Data Set I). (2) The sharp cliff across upper and left side and approaching the diagonal is a sign of a major break between sites. (3) Several contour lines (E's and F's) approach the diagonal between adjacent pottery phases.

4. A series in which there are column and row errors will plot erratically with looping ridges and plateaus. Figure 3-8 shows the ideal of Figure 3-5 with two groups of three sites reversed. Figure 3-9 shows a case in which real data are badly out of order.

5. Discontinuities and/or sharp differences in matrix values will plot as horizontal and vertical cliffs that run up to the diagonal (Fig. 3-8). If this happens it makes the plot harder to read because it breaks the pattern. For example, the contours of Deh Luran pottery (Figs. 3-11 to 3-13) all show the sharp break between Ali Kosh, where the correlations are all less than 17, and Tepe Sabz where the in-site correlations are all greater than 45. In this case the sharp break did not cause much trouble, but if there had been other similar breaks or if this had come elsewhere in the sequence, the plot would have been harder to read.
6. In many sequences there are relatively abrupt changes in value from one column to the next; these may plot as a cliff which approaches but does not necessarily touch the diagonal (Fig. 3-8). In Data Set I, we interpret this behavior as indicating the essential homogeneity of the components within the phases and the differences between phases.

7. If a plot is made of components from a stratified site, the matrix values for the components at the late end of the site are likely to plot as a low profile; those at the early end as a sharply rising profile. Brainerd (1951: 311) attributes this to mixing in the upper part of the site. Our plot of Susiana Black-on-Buff (Data Set Ia) shows this effect (Fig. 3-10).

8. Plots with different contour intervals help point out features that may be missed in a single plot. Plots with many contour intervals are too confusing for easy comprehension and cruder plots, while exposing broad patterns, may overlook the details (Figs. 3-11 to 3-13).

We also reproduce a set of figures to show the various styles of plots that may be produced by the computer. The first gives each contour line of the plot a width proportional to the slope of the surface at each point along the contour; a wide contour line indicates that the matrix values are rising gradually and a narrow line indicates rapid change (Fig. 3-14). A second type of plot gives a line of approximately constant width (Fig. 3-15).
Figure 3-9. Contour plot of correlation matrix in incorrect order (Robinson's data in input order: IIA, IIB, IIC, IA, IB, IIIA, IIIB, IIIC). Note looping ridges. Order of sites in matrix is indicated along left margin of the plot.

A third form of presentation shows a photograph of a contour plot as it appeared on an oscilloscope rather than on paper (Figs. 3-16, 3-18). Another variation is to photograph stereo plots of contours on an oscilloscope, and then view the photographs through a stereo viewer. This renders the contour plot in its true three-dimensional aspect making the peaks, cliffs, and ridges apparent (Figs. 3-17, 3-19). The plots can be viewed through a standard, pocket-sized viewer of the type used to study aerial photographs or they may be viewed without special equipment by placing a piece of cardboard vertically between the two images and bringing them into stereo focus by the eyes alone. A final representation of the topographic characteristics of a contour plot may be obtained by drawing cross sections through various points (Fig. 3-20).

Percentage Plots: A second method of graphically representing the seriation is to plot the percentages of the types in the order given by the program. In this way we see whether the types do conform to a lenticular model of change (Fig. 3-21).

Percentage plots present data in a form that is familiar and usable to archeologists and make it easy to observe the behavior of each type. For example, in one instance, we found that two types of Susiana Black-on-
Figure 3-10. Contour plot of Data Subset Ia (Susiana Black-on-Buff pottery). Order of sites in matrix is indicated along left margin of the plot.

Figure 3-11. Contour plot of Data Set I, (Deh Luran pottery) 10 contours. Order of sites in matrix is indicated along left margin of the plot.
Figure 3-12. Contour plot of Data Set I, (Deh Luran pottery) 8 contours. Order of sites in matrix is indicated along left margin of the plot.

Figure 3-13. Contour plot of Data Set I, (Deh Luran pottery) 6 contours. Order of sites in matrix is indicated along left margin of the plot.
Figure 3-14. Contour plot illustrating technique of making width of lines proportional to slope.

Figure 3-15. Contour plot illustrating technique of making lines of equal width. The "s" occur at the points corresponding to actual matrix entries.
Figure 3-16. Photograph of contour plot produced on an oscilloscope (Robinson's data).

Figure 3-17. Photograph of stereographic contour plot produced on an oscilloscope (Robinson's data).
Figure 3-18. Photograph of contour plot produced on an oscilloscope (Susiana Black-on-Buff pottery).

Figure 3-19. Photograph of stereographic contour plot produced on an oscilloscope (Susiana Black-on-Buff pottery).
Buff pottery had exactly the same distribution. Upon reexamining the typology we found that when working with fragments of pottery, we had classified as a type fragments that we knew came from the bottom of one of two otherwise recognized vessel forms. When we checked the distributions of the vessels we found that one matched the bottoms and the other didn’t; hence, we discovered to which type of vessel the bottoms belonged.
Naturally, if the percentage plots of the types in an ordered matrix do not show the characteristic lenticular pattern, this is a strong indication that the data set is not amenable to seriation and that the final order achieved by the program bears little relation to chronology.

Figure 3-21. Type-Percentage plot of Robinson's data.
CHAPTER IV
COMPARISON OF RESULTS

The Illustrative Data

In this study we used a wide variety of data, both published and unpublished, to test various algorithms and to find suitable examples to illustrate particular points of method or theory. We contemplated using data from several sources but decided that it would facilitate understanding of our results if we relied heavily on data gathered by Rice University expeditions in Iran. These data were collected and studied in a uniform fashion and they are as varied as we need for illustrating our conclusions. It is only incidental that our data come from excavations in Iran; in this paper we are concerned more with the data themselves and the methods of ordering them than we are with interpreting Iranian prehistory.

The data were gathered during three seasons of excavation in the regions of Deh Luran (Khuzistan), and Khorramabad (Luristan), both in west Iran. The largest and most varied set of data comes from Deh Luran.

The Deh Luran Sequence (Data Sets I, II, IV): In brief, the Deh Luran sequence comprises seven phases, each of which is distinguished stratigraphically and six of which are each further subdivided into two or three stratigraphic zones (see Table 4-1). The seven phases were defined after excavations in two sites, Tepe Sabz in which there were four phases, and in Ali Kosh with three phases. In addition, we have a single component from another site, Tepe Musiyan. The sites, all typical near Eastern tells dug in 1963, are visible from one another on a clear day (see Hole, Flannery and Neely 1965).

The relative chronological position of the phases in each site is known through stratigraphy. There may be a chronological gap on the order of some hundreds of years between the Mohammad Jaffar and Sabz Phases but for the present we are not sure of the exact duration of the interval. For the remainder of the sequence, there seems no reason to assume that it was not essentially continuous; there were no obvious stratigraphic gaps of long duration nor has any subsequent study of the artifacts suggested any major breaks.

We might ask, however, what is the actual total length of time represented by the Ali Kosh-Tepe Sabz sequence and what are the relative
TABLE 4-1
THE DEH LURAN STRATIGRAPHIC SEQUENCE

<table>
<thead>
<tr>
<th>Approximate Age-Years B.C.</th>
<th>Site</th>
<th>Phase</th>
<th>Order Name</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>3500-4000</td>
<td>Tepe Sabz</td>
<td>Bayat</td>
<td>BAY1</td>
<td>A-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BAY2</td>
<td>A-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BAY3</td>
<td>A-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MUSE</td>
<td>M-E</td>
</tr>
<tr>
<td>4500-4000</td>
<td>Tepe Sabz</td>
<td>Mehmeh</td>
<td>MEH1</td>
<td>B-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MEH2</td>
<td>B-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MEH3</td>
<td>B-3</td>
</tr>
<tr>
<td>Tepe Musiyan “E”</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000-4500</td>
<td>Tepe Sabz</td>
<td>Khazineh</td>
<td>KHA1</td>
<td>C-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KHA2</td>
<td>C-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KHA3</td>
<td>C-3</td>
</tr>
<tr>
<td>5500-5000</td>
<td>Tepe Sabz</td>
<td>Sabz</td>
<td>SABZ</td>
<td>D</td>
</tr>
<tr>
<td>6000-5600</td>
<td>Ali Kosh</td>
<td>Mohammad Jaffar</td>
<td>MOH1</td>
<td>E-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MOH2</td>
<td>E-2</td>
</tr>
<tr>
<td>6750-6000</td>
<td>Ali Kosh</td>
<td>Ali Kosh</td>
<td>ALI1</td>
<td>F-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ALI2</td>
<td>F-2</td>
</tr>
<tr>
<td>7500-6750</td>
<td>Ali Kosh</td>
<td>Bus Mordeh</td>
<td>BUS1</td>
<td>G-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BUS2</td>
<td>G-2</td>
</tr>
</tbody>
</table>

lengths of time of each of the phases? Unfortunately, radiocarbon dating, the only independent method we have, cannot date the lengths of the phases with very much precision because the range of variation in dates of several samples taken from one phase may be greater than we should expect among samples of different phases. For this reason we shall only mention that the Sabz through Bayat Phases occurred between 5500 and 3800 B.C. and the Bus Mordeh through Mohammad Jaffar Phases occurred between 8000 and 5500 B.C. Clearly with this limited information we cannot say much about rates of change.

The component excavated at nearby Tepe Musiyan, in a very brief test, is included to see how it seriates with reference to the master sequence.

For the seriation based on Data Sets II and IV, we use all 17 components (each stratigraphic zone is a component) but we have only 13 components in the pottery seriation (Data Set I) because the Bus Mordeh and Ali Kosh Phases are preceramic.

Ali Kosh and Tepe Sabz were occupied during a period of great cultural change. The earliest occupants of Ali Kosh were among the first people in the world to experiment with agriculture and stockbreeding to supplement their diet. As time passed, effective agriculture was established and by the time Tepe Sabz was settled, people had discovered and were growing the major hybrids that characterize the area even today. Agriculture had become much more efficient as new techniques, including irrigation, were exploited. In addition, new crafts, such as pottery-making and weaving, and the use of wheels were being developed and the people engaged in wide-
spread if sporadic trade with areas outside Deh Luran. In some respects the people in Deh Luran were isolated but they were not entirely out of contact with the rest of the world nor were they unaffected by the significant developments that were going on about them. For these reasons we may expect that the tools and perhaps other artifacts will vary in response to nonlocal conditions. This will mean that we can expect a good many items to behave erratically in a program that is designed to cope with change in a more closed cultural system.

The Khorramabad Sequence (Data Set III): Additional data come from Pa Sangar, a rockshelter on the outskirts of Khorramabad in west Iran. The deposits in Pa Sangar exhibited no discernible natural layering; hence, we dug them by arbitrary vertical units of ten cm each. For purposes of seriation the tools from each ten cm level were treated as if they had come from separate components.

The Pa Sangar data consist solely of types of flint tools which pertain to the late Baradostian and Zarzian Phases, roughly 30,000-10,000 years before the present. These phases are both prepottery and preagricultural.

In our reconstruction, the people were nomadic hunters living periodically in small caves and rockshelters. As a consequence of repeated occupation of these sites, deposits of cultural material accumulated. During the time in question, there was no great change in the way of life and the people continued to hunt the same animals in the same kind of environment, making their tools out of the same kind of flint.

In some ways the picture is that of small isolated groups who lived in a static environmental situation where changes in the tool types were likely to be of internal inspiration reflecting neither functional differences nor outside cultural influences. Except for the fact that the trends of change parallel those common to the Zagros part of Southwest Asia—hence they indicate some interaction among peoples—there is reason to believe that a nearly ideal cultural situation for seriation existed.

Although we feel that the cultural situation was nearly ideal, we are not so confident we were able to obtain a representative sample of any period. The problems were the lack of natural stratigraphy in Pa Sangar and the possibility that during at least some of the encampments, unusual activities were carried out. In a site where the occupation was continuous one would expect such problems to be minor but this is not necessarily true if people spent only several days at a time in the site. If these unusual activities required either different tools or greatly different proportions of tools from the "typical" assemblage, then the sample would clearly be skewed.

Seriation of Data Sets: In our study of seriation techniques we used each of the methods to obtain orderings for each data set. We then compared these orderings with the original data and tried to determine what factors in
the data themselves caused the various programs to produce the seriations they did.

**Data Set I. Deh Luran Pottery**

This set contains 57 types of pottery discovered in the Deh Luran excavations. These are subdivisions of the seven "types" we distinguish in Hole, Flannery and Neely (in press). The correct stratigraphic sequence for these data is known from our excavations and can be seen in Table 4-1. Each stratigraphic zone in which pottery was found is identified by a letter.

**TABLE 4-2a**

**ORDERINGS FOR DATA SET I, DEH LURAN POTTERY**

(BRAINERD-ROBINSON MATRIX)

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Meighan* 3-Pole</th>
<th>Meighan PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-3</td>
<td>A-2</td>
<td>A-3</td>
<td>A-1</td>
<td>A-3</td>
<td>A-3</td>
</tr>
<tr>
<td>A-3</td>
<td>A-1</td>
<td>A-3</td>
<td>A-1</td>
<td>A-3</td>
<td>A-1</td>
<td>A-2</td>
</tr>
<tr>
<td>M-E</td>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>M-E</td>
<td>B-1</td>
<td>M-E</td>
</tr>
<tr>
<td>B-1</td>
<td>B-2</td>
<td>M-E</td>
<td>B-2</td>
<td>B-2</td>
<td>B-2</td>
<td>B-2</td>
</tr>
<tr>
<td>B-2</td>
<td>M-E</td>
<td>B-2</td>
<td>M-E</td>
<td>B-1</td>
<td>M-E</td>
<td>B-1</td>
</tr>
<tr>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
</tr>
<tr>
<td>C-1</td>
<td>C-1</td>
<td>D</td>
<td>C-1</td>
<td>C-1</td>
<td>C-1</td>
<td>C-3</td>
</tr>
<tr>
<td>C-2</td>
<td>C-2</td>
<td>C-3</td>
<td>C-2</td>
<td>C-2</td>
<td>C-2</td>
<td>C-1</td>
</tr>
<tr>
<td>C-3</td>
<td>C-3</td>
<td>C-1</td>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
</tr>
<tr>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
</tr>
</tbody>
</table>

Norm 244 229 547 229 249 229 460

EC 0 2 7 2 2 2 10

*Result is independent of the type of matrix (see Table 5-1 for types used).

**TABLE 4-2b**

**ORDERINGS FOR DATA SET I, DEH LURAN POTTERY**

(PRESENCE-ABSENCE MATRIX)

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Meighan 3-Pole</th>
<th>Meighan PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>D</td>
<td>A-3</td>
<td>D</td>
<td>A-1</td>
<td>D</td>
<td>A-1</td>
</tr>
<tr>
<td>A-2</td>
<td>A-1</td>
<td>B-1</td>
<td>M-E</td>
<td>A-2</td>
<td>A-1</td>
<td>A-3</td>
</tr>
<tr>
<td>M-E</td>
<td>M-E</td>
<td>B-3</td>
<td>B-1</td>
<td>M-E</td>
<td>M-E</td>
<td>B-1</td>
</tr>
<tr>
<td>B-1</td>
<td>A-3</td>
<td>M-E</td>
<td>B-2</td>
<td>B-2</td>
<td>A-3</td>
<td>A-2</td>
</tr>
<tr>
<td>B-2</td>
<td>B-1</td>
<td>C-1</td>
<td>B-3</td>
<td>B-1</td>
<td>B-1</td>
<td>B-2</td>
</tr>
<tr>
<td>B-3</td>
<td>B-2</td>
<td>A-1</td>
<td>C-2</td>
<td>B-3</td>
<td>B-3</td>
<td>B-2</td>
</tr>
<tr>
<td>C-1</td>
<td>B-3</td>
<td>A-2</td>
<td>C-1</td>
<td>C-1</td>
<td>B-3</td>
<td>C-3</td>
</tr>
<tr>
<td>C-2</td>
<td>C-2</td>
<td>C-3</td>
<td>C-2</td>
<td>C-2</td>
<td>C-2</td>
<td>C-3</td>
</tr>
<tr>
<td>C-3</td>
<td>C-1</td>
<td>C-3</td>
<td>A-2</td>
<td>C-3</td>
<td>C-1</td>
<td>A-1</td>
</tr>
<tr>
<td>D</td>
<td>C-3</td>
<td>D</td>
<td>A-1</td>
<td>D</td>
<td>C-3</td>
<td>C-3</td>
</tr>
<tr>
<td>E-1</td>
<td>E-1</td>
<td>E-2</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
</tr>
<tr>
<td>E-2</td>
<td>E-2</td>
<td>E-1</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
</tr>
</tbody>
</table>

Norm 145 113 203 128 163 113 192

EC 0 11 11 23 2 11 23

*Result is independent of the type of matrix (see Table 5-1 for types used).
and number which run from A-1 for the latest to E-2 for the oldest. (The raw data are given in Appendix A.)

As Tables 4-2a to 4-5b show, the results of running the different algorithms against the two matrix forms diverge widely. In each case, however, Permutation Search (with or without preordering) arrived at the lowest norm and, except in one case using a Presence-Absence matrix, at the lowest Error Coefficient. This pattern of results holds true for the remainder of the data sets and can be explained by the characteristics of the algorithms with respect to input order and to ambiguous data (see explanation in Chapters III and V).

**TABLE 4-3a**

**ORDERINGS FOR DATA SUBSET Ia, SUSIANA BLACK-ON-BUFF POTTERY (BRAINERD-ROBINSON MATRIX)**

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-1</td>
<td>A-1</td>
<td>D</td>
<td>A-2</td>
</tr>
<tr>
<td>A-2</td>
<td>A-2</td>
<td>A-2</td>
<td>C-3</td>
<td>A-1</td>
</tr>
<tr>
<td>A-3</td>
<td>A-3</td>
<td>A-3</td>
<td>A-3*</td>
<td>A-3</td>
</tr>
<tr>
<td>M-E</td>
<td>B-1</td>
<td>B-1</td>
<td>A-1</td>
<td>B-1</td>
</tr>
<tr>
<td>B-1</td>
<td>B-2</td>
<td>M-E</td>
<td>A-2</td>
<td>B-2</td>
</tr>
<tr>
<td>B-2</td>
<td>M-E</td>
<td>B-2</td>
<td>B-1</td>
<td>M-E</td>
</tr>
<tr>
<td>B-3</td>
<td>B-3</td>
<td>D</td>
<td>B-2</td>
<td>B-3</td>
</tr>
<tr>
<td>C-1</td>
<td>C-3</td>
<td>C-3</td>
<td>M-E</td>
<td>D</td>
</tr>
<tr>
<td>C-2</td>
<td>C-2</td>
<td>M-E</td>
<td>D</td>
<td>C-3</td>
</tr>
<tr>
<td>C-3</td>
<td>C-3</td>
<td>B-3</td>
<td>B-3</td>
<td>C-3</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>C-2</td>
<td>C-2</td>
<td>C-2</td>
</tr>
</tbody>
</table>

Norm 323×

EC 0

DB found a norm of 713 but did not stay with it. EC 12 on 713.

* Error here is 80.6; next largest inter-column error is 34.2.

× Error coefficient is 0, but norm is lower for Permutation Search because M-E is not counted in Error Coefficient.

**TABLE 4-3b**

**ORDERINGS FOR DATA SUBSET Ia, SUSIANA BLACK-ON-BUFF POTTERY (PRESENCE-ABSENCE MATRIX)**

<table>
<thead>
<tr>
<th>Input Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-1</td>
<td>B-1</td>
<td>A-1</td>
<td>D</td>
</tr>
<tr>
<td>A-2</td>
<td>A-2</td>
<td>A-3</td>
<td>A-2</td>
<td>A-1</td>
</tr>
<tr>
<td>A-3</td>
<td>M-E</td>
<td>A-1</td>
<td>A-1</td>
<td>A-2</td>
</tr>
<tr>
<td>M-E</td>
<td>B-1</td>
<td>A-2</td>
<td>A-3</td>
<td>M-E</td>
</tr>
<tr>
<td>B-1</td>
<td>A-3</td>
<td>M-E</td>
<td>B-2</td>
<td>B-1</td>
</tr>
<tr>
<td>B-2</td>
<td>B-2</td>
<td>B-2</td>
<td>B-2</td>
<td>B-1</td>
</tr>
<tr>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
</tr>
<tr>
<td>C-1</td>
<td>C-1</td>
<td>C-1</td>
<td>C-1</td>
<td>C-1</td>
</tr>
<tr>
<td>C-2</td>
<td>C-2</td>
<td>D</td>
<td>C-2</td>
<td>C-2</td>
</tr>
<tr>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>C-2</td>
<td>D</td>
<td>C-3</td>
</tr>
</tbody>
</table>

Norm 31

EC 0

19 83 19 25

2 5 2 8
Not all of our results can be explained by the mechanics of ordering. For this we must turn to the data themselves. In Data Set I we are especially interested in learning what factors caused A-1, A-2, and A-3 to appear in inverse order in the best seriations.

The 57 types of pottery which we distinguish here can be regrouped into the seven types in our site report which are distinguished only by color, decoration, or surface treatment. If we run one of these, Subset 1a (Susiana Black-on-Buff), we get the correct stratigraphic order with Permutation Search alone but the order gets garbled when we run Permutation Search on the input order given by the Dempsey-Baumhoff run (Table 4-3a). Therefore, Data Subset 1a showed that it was not Susiana Black-on-Buff that was causing the reversal in order of A-1, -2, -3 when the full set was run.

The performance of Permutation Search on the input order given by Dempsey-Baumhoff can be explained by reference to the matrix of the data subset. It shows that component D varies greatly from the remainder of the components; the error between C-3 and A-3 is 80.6 but the next largest error between columns is 34.2. In this case the Permutation Search algorithm took item D out of the order given it by Dempsey-Baumhoff and tried it at the left end of the matrix. Since this resulted in a lower norm, it left D in its new position and then tried C-3 to achieve a better matrix. This again improved the matrix and without exhaustive search, Permutation Search concluded that it had the best possible result. Mechanically this is understandable but we should like to see what kind of data caused the problem.

The distribution of some types is restricted entirely or almost entirely to component D; for other types the closest resemblance is found in C-3, which is not surprising since D and C-3 are adjacent stratigraphically. One other type, which comprises more than a third of all the sherds, changes dramatically in frequency between C-3 and C-2. These factors tend to split D from C-3 and from the other components, wherever they may be placed, and the chances for misplacement are enhanced when a search for the best fit is not exhaustive.

To continue our attempt to understand the behavior of components A-1, -2, -3, we tried another subset, 1b (Susiana Buff). Runs of this showed the reversal we had found on the full data set but the orderings in general were worse than we had obtained on previous sets (Table 4-4a). In part this reflects the nature of the Buff pottery. The percentages of certain types depended on our finding sherds that had no painting on them. During periods when it was customary to paint the greater portion of the surface of a vessel, the percentage of Buff to painted pottery dropped, and the reverse was also true. This accounts for some of the variability in the percentages of the types from one component to the next. The Buff types vary both with
respects to time and with respect to the Black-on-Buff. The same effect was noticed by Dixon (1956:10). As it turns out, the Buff pottery has a bimodal distribution, hitting peaks in frequency at the temporal extremes of our sequence. If the Susiana Buff had not varied at all with time (some of its constituent types do), it would not have seriated as well as it did. The bulk of the sherds counted were from the bodies of vessels that could have been either plain or painted; there are relatively few types that can be invariably distinguished on the basis of body sherds.

The final subset pertinent to the question of the sequence of components A-1, -2, and -3 is Bayat Red, a pottery in which three types occur in the upper four components. Seriating these 70 sherds results in the re-
versed order again (Tables 4-5a and 4-5b) so it is plain that with the exception of the Black-on-Buff, all data subsets contributed to the incorrect order attained with the full set.

**TABLE 4-5a**

**ORDERINGS FOR DATA SUBSET IC, BAYAT RED POTTERY**

<table>
<thead>
<tr>
<th>Input Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-3</td>
<td>A-3</td>
<td>A-3</td>
<td>A-3</td>
</tr>
<tr>
<td>A-3</td>
<td>A-1</td>
<td>A-1</td>
<td>A-1</td>
<td>A-1</td>
</tr>
<tr>
<td>M-E</td>
<td>M-E</td>
<td>M-E</td>
<td>M-E</td>
<td>B-1</td>
</tr>
<tr>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>M-E</td>
</tr>
</tbody>
</table>

Norm 76
EC 0

**TABLE 4-5b**

**ORDERINGS FOR DATA SUBSET IC, BAYAT RED POTTERY**

<table>
<thead>
<tr>
<th>Input Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>PS DB</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-3</td>
<td>A-3</td>
<td>A-3</td>
<td>A-3</td>
</tr>
<tr>
<td>A-3</td>
<td>A-1</td>
<td>A-1</td>
<td>A-1</td>
<td>A-1</td>
</tr>
<tr>
<td>M-E</td>
<td>M-E</td>
<td>M-E</td>
<td>M-E</td>
<td>B-1</td>
</tr>
<tr>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>M-E</td>
</tr>
</tbody>
</table>

Norm 5
EC 0

There remains the possibility of mixing— a situation anticipated by Brain- erd in the upper levels of a site like Tepe Sabz— to account for the reversal in order, but this cannot be checked until we have excavated an isolated component in which there would be no chance of mixing.

The placing of M-E by the various methods has some bearing on the question of mixing, however. Whenever a Brainerd-Robinson matrix is used on Data Sets I and Ia, B-2 is always adjacent to M-E. In most cases the component on the other side of M-E is one of the B’s. That is, for these data sets, M-E is most like the B’s. Subset Ib, however, shows M-E aligned with C and D, but Subset Ic shows M-E between B-1 and one of the A’s.

Turning to the Presence-Absence matrix, we find that although there is more variation in the results, M-E usually seriates between a B and an A, no matter which subset of data is used.

These results suggest that M-E contains elements of both A and B and is not a “pure” component. We should expect to find this the rule for components in any relatively continuous sequence. We conclude therefore that the sort of “mixing” we find in any component could be the natural effect of types conforming to the lenticular pattern of change.
The results of seriation seem to show that the phases in Tepe Sabz were correctly identified during our preseriation analyses. We interpret the clumping of components in the various phases as an indication of the essential homogeneity of the phases.

Whenever there is any mixing of deposits—and this is almost a certainty in stratified sites—a Presence-Absence matrix will find it more difficult to arrive at the correct seriation. This was borne out by our findings; we invariably achieved worse results than on a Brainerd-Robinson matrix. Whatever the merits of using Presence-Absence as a criterion, we should not forget that in a finely divided sequence, change in the relative percentages of types is as important a factor as is mere presence or absence. Therefore one would expect the presence-absence runs to give consistently poorer results. Tables 4-2b, -3b, -4b bear this out.

The results of seriating the Deh Luran pottery are eminently satisfactory and fully vindicate the use of techniques of mechanical seriation. The best results were obtained with all methods when they were run on the full data set using the Brainerd-Robinson matrix. Even when the orderings were slightly out of line with the stratigraphic order, all the methods kept the zones from each phase together.

Data Set II. Deh Luran Flint

Since the pottery seriated well we were encouraged to try runs with other kinds of artifacts from the same sequence. We chose 33 types of chipped stone tools which were present in some or all of the 17 components. The results of some trial runs suggested that flint would not seriate with the same dependability as pottery. Permutation Search separated the components in the two sites but it did not discriminate their order within the sites very well. Therefore, we tried to set up a subset of flint that would seriate better. After comparing the results of our trial runs with raw data it seemed that two factors affected the seriation: some types swamped the matrix because of their extraordinary abundance, and other types confused the issue because they did not vary with respect to time. Consequently we took advantage of knowing the stratigraphy and attempted to construct a subset of data that would satisfy our theoretical conditions. We picked types that we knew were restricted in time or which varied in frequency with time. Further, we tried to choose types that had neither unusual abundance nor scarcity. Finally, we added to our list some cores which, although not tools, did seem to contribute some information to our seriation. We did this after we eliminated plain blades; their abundance was too great and we replaced them with cores on the grounds that they measured the same things but, for our data, in more reasonable proportions. After manipulating the data in this way we arrived at 13 types for our final matrix (Appendix A). These types represented some lumping of previously small categories and the addi-
tion of several types of cores. The results of the seriations on these pre-judged data are in Tables 4-6a and 4-6b.

### TABLE 4-6a

**ORDERINGS FOR DATA SET II, ALI KOSH-TEPE SABZ FLINT**  
(BRAINERD-ROBINSON MATRIX)

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Meighan 3-Pole</th>
<th>Meighan PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>M-E</td>
<td>C-2</td>
<td>D</td>
<td>C-2</td>
<td>D</td>
<td>A-2</td>
</tr>
<tr>
<td>A-2</td>
<td>A-2</td>
<td>D</td>
<td>C-1</td>
<td>M-E</td>
<td>C-1</td>
<td>A-3</td>
</tr>
<tr>
<td>A-3</td>
<td>A-3</td>
<td>C-1</td>
<td>C-2</td>
<td>A-3</td>
<td>C-2</td>
<td>C-3</td>
</tr>
<tr>
<td>M-E</td>
<td>A-1</td>
<td>C-3</td>
<td>C-3</td>
<td>B-1</td>
<td>C-3</td>
<td>C-2</td>
</tr>
<tr>
<td>B-1</td>
<td>B-1</td>
<td>A-2</td>
<td>B-3</td>
<td>A-1</td>
<td>B-3</td>
<td>D</td>
</tr>
<tr>
<td>B-2</td>
<td>B-2</td>
<td>B-3</td>
<td>B-1</td>
<td>A-2</td>
<td>B-1</td>
<td>A-1</td>
</tr>
<tr>
<td>B-3</td>
<td>C-3</td>
<td>B-1</td>
<td>A-1</td>
<td>C-3</td>
<td>A-1</td>
<td>A-3</td>
</tr>
<tr>
<td>C-1</td>
<td>C-2</td>
<td>B-2</td>
<td>A-3</td>
<td>C-1</td>
<td>A-3</td>
<td>B-1</td>
</tr>
<tr>
<td>C-2</td>
<td>D</td>
<td>M-E</td>
<td>A-2</td>
<td>B-2</td>
<td>A-2</td>
<td>B-3</td>
</tr>
<tr>
<td>C-3</td>
<td>C-1</td>
<td>A-1</td>
<td>M-E</td>
<td>B-3</td>
<td>M-E</td>
<td>B-2</td>
</tr>
<tr>
<td>D</td>
<td>B-3</td>
<td>A-3</td>
<td>B-2</td>
<td>D</td>
<td>B-2</td>
<td>G-2</td>
</tr>
<tr>
<td>E-1</td>
<td>F-1</td>
<td>F-1</td>
<td>F-1</td>
<td>F-2</td>
<td>F-1</td>
<td>G-1</td>
</tr>
<tr>
<td>E-2</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>F-1</td>
</tr>
<tr>
<td>F-1</td>
<td>E-2</td>
<td>G-1</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>F-1</td>
</tr>
<tr>
<td>F-2</td>
<td>F-2</td>
<td>G-2</td>
<td>F-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-1</td>
</tr>
<tr>
<td>G-1</td>
<td>G-1</td>
<td>F-2</td>
<td>G-1</td>
<td>G-1</td>
<td>G-1</td>
<td>E-2</td>
</tr>
<tr>
<td>G-2</td>
<td>G-2</td>
<td>G-2</td>
<td>G-2</td>
<td>G-2</td>
<td>M-E</td>
<td></td>
</tr>
</tbody>
</table>

| Norm | 1714 | 734 | 2157 | 866 | 2095 | 866 | 2228 |
| EC   | 0    | 17  | 33   | 19  | 23   | 19  | 27   |

### TABLE 4-6b

**ORDERINGS FOR DATA SET II, ALI KOSH-TEPE SABZ FLINT**  
(PRESENCE-ABSENCE MATRIX)

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Meighan 3-Pole</th>
<th>Meighan PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>M-E</td>
<td>D</td>
<td>M-E</td>
<td>C-2</td>
<td>M-E</td>
<td>M-E</td>
</tr>
<tr>
<td>A-3</td>
<td>C-2</td>
<td>C-2</td>
<td>C-2</td>
<td>A-3</td>
<td>C-2</td>
<td>C-2</td>
</tr>
<tr>
<td>M-E</td>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
</tr>
<tr>
<td>B-1</td>
<td>A-1</td>
<td>A-1</td>
<td>A-1</td>
<td>A-1</td>
<td>C-3</td>
<td>A-1</td>
</tr>
<tr>
<td>B-2</td>
<td>B-3</td>
<td>A-1</td>
<td>B-3</td>
<td>A-2</td>
<td>B-2</td>
<td>B-2</td>
</tr>
<tr>
<td>B-3</td>
<td>C-3</td>
<td>A-3</td>
<td>C-1</td>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
</tr>
<tr>
<td>C-1</td>
<td>A-3</td>
<td>B-3</td>
<td>A-3</td>
<td>B-3</td>
<td>B-3</td>
<td>A-3</td>
</tr>
<tr>
<td>C-2</td>
<td>B-2</td>
<td>C-3</td>
<td>C-3</td>
<td>B-3</td>
<td>A-3</td>
<td>A-3</td>
</tr>
<tr>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
<td>C-3</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>F-2</td>
<td>F-2</td>
<td>G-2</td>
</tr>
<tr>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
<td>F-2</td>
<td>F-1</td>
<td>G-1</td>
</tr>
<tr>
<td>E-2</td>
<td>F-2</td>
<td>G-2</td>
<td>G-2</td>
<td>E-2</td>
<td>F-2</td>
<td>F-2</td>
</tr>
<tr>
<td>F-1</td>
<td>F-1</td>
<td>G-1</td>
<td>G-1</td>
<td>E-1</td>
<td>G-2</td>
<td>F-2</td>
</tr>
<tr>
<td>F-2</td>
<td>F-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
<td>F-1</td>
</tr>
<tr>
<td>G-1</td>
<td>G-1</td>
<td>F-2</td>
<td>F-2</td>
<td>G-1</td>
<td>G-1</td>
<td>E-2</td>
</tr>
<tr>
<td>G-2</td>
<td>G-2</td>
<td>F-1</td>
<td>F-1</td>
<td>E-2</td>
<td>E-1</td>
<td>E-1</td>
</tr>
</tbody>
</table>

| Norm | 127  | 49  | 58   | 48  | 101  | 53  | 69   |
| EC   | 0    | 25  | 35   | 28  | 23   | 40  | 30   |

Permutation Search again found the best results, but we found a curious thing. Although the components in Ali Kosh and Tepe Sabz were kept in
reasonably good order internally, the sequence taken as a whole is quite misleading. In some orderings (although not on the best Permutation Search order) the upper portion of Ali Kosh was placed adjacent to the upper portion of Tepe Sabz (that is, the E's or F's were put next to the A's or B's). Thus, although the orders within each site were acceptable, they were reversed with respect to one another.

The cause of the erratic behavior is easy to see in the percentages of the types. When the types were preselected they were picked by eye from raw data on a master chart that plotted the stratigraphic occurrence of the artifacts. After the percentages were printed on the computer it was obvious that most of the types did not conform to the model of lenticular change. As a consequence, there is no way our algorithms could consistently come out with the stratigraphic order.

Another problem worth mentioning is that there are relatively few flints in the Tepe Sabz components and consequently some types were probably absent accidentally from our samples. In Tepe Sabz, a small change in quantity could readily affect the matrix. The problem of accidental omissions is probably more severe in Tepe Sabz than in Ali Kosh.

### TABLE 4-7

**ORDERINGS FOR DATA SET III, PA SANGAR Flint, 33 TYPES**  
(BRAINERD-ROBINSON AND PRESENCE-ABSENCE MATRICES),  
AND FOR COMPONENTS GROUPED BY 40 CM UNITS

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>BR Matrix Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>± Matrix Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>BR Matrix Perm Search Lumped Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>34</td>
<td>34</td>
<td>23</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>23</td>
<td>56</td>
<td>56</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>01</td>
<td>12</td>
<td>12</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>12</td>
<td>78</td>
<td>67</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>56</td>
<td>67</td>
<td>45</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>78</td>
<td>23</td>
<td>89</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>67</td>
<td>45</td>
<td>78</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>45</td>
<td>01</td>
<td>134</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>89</td>
<td>89</td>
<td>112</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>90</td>
<td>90</td>
<td>156</td>
<td>134</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>134</td>
<td>134</td>
<td>101</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>123</td>
<td>123</td>
<td>123</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>101</td>
<td>101</td>
<td>145</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>112</td>
<td>167</td>
<td>189</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>145</td>
<td>145</td>
<td>167</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>189</td>
<td>112</td>
<td>90</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>156</td>
<td>189</td>
<td>178</td>
<td>145</td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>167</td>
<td>178</td>
<td>201</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>189</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>178</td>
<td>156</td>
<td>01</td>
<td>201</td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>201</td>
<td>201</td>
<td>34</td>
<td>190</td>
<td>F (201)</td>
</tr>
<tr>
<td>Norm 2180</td>
<td>689</td>
<td>1414</td>
<td>152</td>
<td>289</td>
<td>0</td>
</tr>
<tr>
<td>EC 0</td>
<td>25</td>
<td>49</td>
<td>67</td>
<td>70</td>
<td>0</td>
</tr>
</tbody>
</table>


Data Set III. Pa Sangar Flint

The poor results we obtained with the Ali Kosh-Tepe Sabz flint tools led us to question whether such types would ever be good for seriation. If they were inherently ill-suited for the purpose, it would be of great concern to archeologists working with paleolithic sites where stone tools are virtually the only artifacts preserved. The fact is, flints have been seriated by several methods with apparent good results for many years (e.g., Bordes 1952, Jelinek 1962). To test the notion, however, we tried a data set that had been excavated in Pa Sangar, a paleolithic site in the vicinity of Khorramabad, west Iran during the summer of 1965 (Appendix A).

The first runs of these data were made via airmail between Iran and Rice University, a few days after the site had been dug and the flints subjected to a preliminary study and count of types. In the initial attempt we used data derived from excavation unit "Pa," one of three at the site (Table 4-7). Although the results did not duplicate the stratigraphic order, they did seriate fairly well using Permutation Search yielding an Error Coefficient of 25 and displacing no component more than three places from its actual position in the ground.

Figure 4-1. Contour plot of Data Set III (Pa Sangar flint). Order of grouped data is indicated along left margin of the plot.
In seeking an explanation for the unimpressive seriation, we examined the values of all the entries in the Brainerd-Robinson matrix. They are all high; only about 5% of them are less than 100, and only about .5% are less than 80 (Table 3-1, Figure 3-1). Since the possible maximum value is 200 the differences among components in the matrices are relatively slight; therefore, discriminating among them to obtain an order is relatively difficult.

Our first reaction to the Pa Sangar results was that the errors had resulted from the stratigraphic cuts being too thin. The ten cm cuts were not "time-units" in the same way that the components in the pottery seriation were. We dismiss an alternative explanation that the deposits were badly mixed because there is no evidence for extensive burrowing or quarrying and because the first seriation was reasonably accurate.

To test the hypothesis that the archeological cuts discriminated too finely, we lumped the components into successive groups of four (i.e., 40 cm units). When this was done, the order came out as expected stratigraphically and does show that there is order in the data (Table 4-7, Fig. 4-1). Incidentally, Figure 4-1 shows a contour plot of a nearly perfect matrix.

**TABLE 4-8a**

**ORDERINGS FOR DATA SUBSET III, PA SANGAR FLINT, 11 TYPES**

(BRAINERD-ROBINSON MATRIX)

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Meighan 3-Pole</th>
<th>Meighan PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>34</td>
<td>45</td>
<td>45</td>
<td>12</td>
<td>45</td>
<td>34</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>34</td>
<td>34</td>
<td>23</td>
<td>34</td>
<td>45</td>
</tr>
<tr>
<td>23</td>
<td>67</td>
<td>23</td>
<td>12</td>
<td>34</td>
<td>12</td>
<td>23</td>
</tr>
<tr>
<td>34</td>
<td>23</td>
<td>12</td>
<td>23</td>
<td>45</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>45</td>
<td>12</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>01</td>
</tr>
<tr>
<td>56</td>
<td>01</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>78</td>
</tr>
<tr>
<td>67</td>
<td>56</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>67</td>
</tr>
<tr>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>56</td>
</tr>
<tr>
<td>89</td>
<td>123</td>
<td>123</td>
<td>123</td>
<td>167</td>
<td>123</td>
<td>190</td>
</tr>
<tr>
<td>90</td>
<td>167</td>
<td>167</td>
<td>167</td>
<td>167</td>
<td>167</td>
<td>189</td>
</tr>
<tr>
<td>101</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>89</td>
<td>167</td>
</tr>
<tr>
<td>112</td>
<td>112</td>
<td>112</td>
<td>112</td>
<td>112</td>
<td>112</td>
<td>123</td>
</tr>
<tr>
<td>123</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>134</td>
<td>101</td>
<td>145</td>
</tr>
<tr>
<td>134</td>
<td>145</td>
<td>134</td>
<td>145</td>
<td>101</td>
<td>145</td>
<td>112</td>
</tr>
<tr>
<td>145</td>
<td>134</td>
<td>190</td>
<td>134</td>
<td>90</td>
<td>134</td>
<td>101</td>
</tr>
<tr>
<td>156</td>
<td>90</td>
<td>145</td>
<td>90</td>
<td>145</td>
<td>90</td>
<td>134</td>
</tr>
<tr>
<td>167</td>
<td>190</td>
<td>90</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>89</td>
</tr>
<tr>
<td>178</td>
<td>189</td>
<td>189</td>
<td>189</td>
<td>201</td>
<td>189</td>
<td>90</td>
</tr>
<tr>
<td>189</td>
<td>201</td>
<td>201</td>
<td>201</td>
<td>189</td>
<td>201</td>
<td>201</td>
</tr>
<tr>
<td>190</td>
<td>156</td>
<td>178</td>
<td>156</td>
<td>156</td>
<td>156</td>
<td>156</td>
</tr>
<tr>
<td>201</td>
<td>178</td>
<td>212</td>
<td>178</td>
<td>178</td>
<td>178</td>
<td>178</td>
</tr>
<tr>
<td>212</td>
<td>212</td>
<td>156</td>
<td>212</td>
<td>212</td>
<td>212</td>
<td>212</td>
</tr>
<tr>
<td>Norm 2180</td>
<td>389</td>
<td>1907</td>
<td>426</td>
<td>562</td>
<td>426</td>
<td>1272</td>
</tr>
<tr>
<td>EC 0</td>
<td>49</td>
<td>64</td>
<td>54</td>
<td>45</td>
<td>54</td>
<td>51</td>
</tr>
</tbody>
</table>


Following the final study of the flint from Pa Sangar, we attempted to preselect those types that had the most potential for seriation. As a result of this we decided upon eleven types which represented a consolidation of some of the original types and the omission of some others. Cores were not included in this data set (Tables 4-8a, 4-8b). In the final seriations the Permutation Search method gave the lowest norm on the Brainerd-Robinson matrix but it had an Error Coefficient of 49, far worse than the results on the unaltered set. In this instance our attempt to improve the data simply intensified the problems that we had already experienced.

We did not make further runs manipulating various Pa Sangar data sets because we felt we understood what had happened. In short, the stratigraphic cuts were more finely divided than the “time-units” as these are expressed in changes in artifacts through time.

### TABLE 4-8b

**Orderings for Data Subset III, Pa Sangar Flint, 11 Types**

(Presence-Absence Matrix)

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Meighan 3-Pole</th>
<th>Meighan PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>01</td>
<td>23</td>
<td>23</td>
<td>12</td>
<td>90</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>34</td>
<td>34</td>
<td>23</td>
<td>67</td>
<td>01</td>
</tr>
<tr>
<td>23</td>
<td>23</td>
<td>01</td>
<td>01</td>
<td>34</td>
<td>23</td>
<td>56</td>
</tr>
<tr>
<td>34</td>
<td>34</td>
<td>12</td>
<td>12</td>
<td>45</td>
<td>34</td>
<td>45</td>
</tr>
<tr>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>67</td>
<td>01</td>
<td>123</td>
</tr>
<tr>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>12</td>
<td>101</td>
</tr>
<tr>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>01</td>
<td>56</td>
<td>34</td>
</tr>
<tr>
<td>78</td>
<td>90</td>
<td>78</td>
<td>90</td>
<td>78</td>
<td>45</td>
<td>23</td>
</tr>
<tr>
<td>89</td>
<td>89</td>
<td>112</td>
<td>89</td>
<td>167</td>
<td>78</td>
<td>67</td>
</tr>
<tr>
<td>90</td>
<td>190</td>
<td>90</td>
<td>112</td>
<td>123</td>
<td>134</td>
<td>112</td>
</tr>
<tr>
<td>101</td>
<td>101</td>
<td>89</td>
<td>78</td>
<td>89</td>
<td>145</td>
<td>90</td>
</tr>
<tr>
<td>112</td>
<td>112</td>
<td>145</td>
<td>145</td>
<td>112</td>
<td>112</td>
<td>78</td>
</tr>
<tr>
<td>123</td>
<td>78</td>
<td>134</td>
<td>134</td>
<td>134</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>134</td>
<td>134</td>
<td>156</td>
<td>101</td>
<td>101</td>
<td>190</td>
<td>156</td>
</tr>
<tr>
<td>145</td>
<td>145</td>
<td>167</td>
<td>167</td>
<td>90</td>
<td>189</td>
<td>190</td>
</tr>
<tr>
<td>156</td>
<td>189</td>
<td>189</td>
<td>189</td>
<td>145</td>
<td>101</td>
<td>189</td>
</tr>
<tr>
<td>167</td>
<td>167</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>123</td>
<td>167</td>
</tr>
<tr>
<td>178</td>
<td>123</td>
<td>123</td>
<td>123</td>
<td>201</td>
<td>167</td>
<td>145</td>
</tr>
<tr>
<td>189</td>
<td>156</td>
<td>101</td>
<td>156</td>
<td>189</td>
<td>156</td>
<td>134</td>
</tr>
<tr>
<td>190</td>
<td>178</td>
<td>178</td>
<td>178</td>
<td>156</td>
<td>178</td>
<td>212</td>
</tr>
<tr>
<td>201</td>
<td>201</td>
<td>201</td>
<td>201</td>
<td>178</td>
<td>212</td>
<td>201</td>
</tr>
<tr>
<td>212</td>
<td>212</td>
<td>212</td>
<td>212</td>
<td>212</td>
<td>201</td>
<td>178</td>
</tr>
</tbody>
</table>

| Norm | 106          | 36               | 45    | 144           | 50         | 114    |
| EC   | 0            | 40               | 30    | 45            | 46         | 47     |

**Data Set IV, 110 Types**

Our final data set was chosen to test the effects of combining many varieties of data. We considered simultaneously all the remaining artifacts from Ali Kosh and Tepe Sabz which occurred in one or more component. This data set excludes the flint and pottery but it does include 110 types of such
artifacts as spindle whorls, chipped sherds, miscellaneous ceramic artifacts, grinding stones, pestles, pounders, other stone tools, figurines, objects of unfired clay, matting, basketry, textiles, bone tools, and ornaments. In other words, we lumped all the data we could (Appendix A).

The results are not bad but they are not as good as we attained using the pottery (Tables 4-9a, 4-9b). Permutation Search did the best job with both forms of matrix. The Error Coefficient is lower for the Presence-Absence matrix but the results on the Brainerd-Robinson matrix seem better from an archeological viewpoint because they kept the material from each phase together. In the Presence-Absence matrix, the ordering of the upper portion of Tepe Sabz, Phases A and B, was mixed.

Including all sorts of artifacts in a data set maximizes the chance that stratigraphic mixing will influence the results by muddying the distinctions. That such mixing did not have more severe consequences than it did is probably due to the fact that a relatively few of the types, by virtue of their abundance, exercised a disproportionate beneficial weight in the Brainerd-Robinson matrix. If some of the data are well conditioned for seriation, this form of matrix gains an advantage. With the Presence-Absence matrix, Phases A and B were mixed and G was placed next to C in two of the orderings, a result that is wholly unacceptable archeologically.

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-2</td>
<td>B-1</td>
<td>A-2</td>
<td>M-E</td>
</tr>
<tr>
<td>A-2</td>
<td>A-3</td>
<td>M-E</td>
<td>A-3</td>
<td>D</td>
</tr>
<tr>
<td>A-3</td>
<td>A-1</td>
<td>B-2</td>
<td>A-1</td>
<td>C-3</td>
</tr>
<tr>
<td>M-E</td>
<td>M-E</td>
<td>A-3</td>
<td>M-E</td>
<td>C-2</td>
</tr>
<tr>
<td>B-1</td>
<td>B-2</td>
<td>B-3</td>
<td>B-2</td>
<td>C-1</td>
</tr>
<tr>
<td>B-2</td>
<td>B-1</td>
<td>A-1</td>
<td>B-1</td>
<td>B-3</td>
</tr>
<tr>
<td>B-3</td>
<td>B-3</td>
<td>A-2</td>
<td>B-3</td>
<td>B-2</td>
</tr>
<tr>
<td>C-1</td>
<td>C-1</td>
<td>C-1</td>
<td>C-1</td>
<td>B-1</td>
</tr>
<tr>
<td>C-2</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>A-3</td>
</tr>
<tr>
<td>C-3</td>
<td>C-3</td>
<td>C-2</td>
<td>C-3</td>
<td>A-1</td>
</tr>
<tr>
<td>D</td>
<td>C-2</td>
<td>C-3</td>
<td>C-2</td>
<td>A-2</td>
</tr>
<tr>
<td>E-1</td>
<td>F-1</td>
<td>F-1</td>
<td>F-1</td>
<td>F-1</td>
</tr>
<tr>
<td>E-2</td>
<td>E-1</td>
<td>G-1</td>
<td>E-1</td>
<td>E-2</td>
</tr>
<tr>
<td>F-1</td>
<td>E-2</td>
<td>F-2</td>
<td>E-2</td>
<td>E-1</td>
</tr>
<tr>
<td>F-2</td>
<td>F-2</td>
<td>E-1</td>
<td>F-2</td>
<td>F-2</td>
</tr>
<tr>
<td>G-1</td>
<td>G-2</td>
<td>E-2</td>
<td>G-2</td>
<td>G-2</td>
</tr>
<tr>
<td>G-2</td>
<td>G-1</td>
<td>G-2</td>
<td>G-1</td>
<td>G-1</td>
</tr>
<tr>
<td>Norm</td>
<td>1178</td>
<td>721</td>
<td>721</td>
<td>1215</td>
</tr>
<tr>
<td>EC</td>
<td>0</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>

TABLE 4-9a
ORDERINGS FOR DATA SET IV, 110 TYPES (BRAINERD-ROBINSON MATRIX)
<table>
<thead>
<tr>
<th>Initial Order</th>
<th>Perm Search</th>
<th>Dempsey-Baumhoff</th>
<th>DB PS</th>
<th>Ascher</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-3</td>
<td>M-E</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>A-2</td>
<td>B-1</td>
<td>A-2</td>
<td>A-3</td>
<td>B-1</td>
</tr>
<tr>
<td>A-3</td>
<td>B-2</td>
<td>B-1</td>
<td>B-1</td>
<td>C-2</td>
</tr>
<tr>
<td>M-E</td>
<td>A-2</td>
<td>B-2</td>
<td>B-2</td>
<td>B-3</td>
</tr>
<tr>
<td>B-1</td>
<td>M-E</td>
<td>C-2</td>
<td>A-2</td>
<td>B-2</td>
</tr>
<tr>
<td>B-2</td>
<td>A-1</td>
<td>A-3</td>
<td>M-E</td>
<td>M-E</td>
</tr>
<tr>
<td>B-3</td>
<td>B-3</td>
<td>B-3</td>
<td>A-1</td>
<td>A-3</td>
</tr>
<tr>
<td>C-1</td>
<td>C-1</td>
<td>A-1</td>
<td>B-3</td>
<td>A-2</td>
</tr>
<tr>
<td>C-2</td>
<td>C-2</td>
<td>C-1</td>
<td>C-1</td>
<td>C-1</td>
</tr>
<tr>
<td>C-3</td>
<td>C-3</td>
<td>D</td>
<td>C-2</td>
<td>C-3</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>C-3</td>
<td>C-3</td>
<td>A-1</td>
</tr>
<tr>
<td>E-1</td>
<td>E-1</td>
<td>G-1</td>
<td>G-1</td>
<td>G-1</td>
</tr>
<tr>
<td>E-2</td>
<td>E-2</td>
<td>G-2</td>
<td>G-2</td>
<td>G-2</td>
</tr>
<tr>
<td>F-1</td>
<td>F-1</td>
<td>F-1</td>
<td>F-1</td>
<td>F-2</td>
</tr>
<tr>
<td>F-2</td>
<td>F-2</td>
<td>E-1</td>
<td>E-2</td>
<td>E-2</td>
</tr>
<tr>
<td>G-1</td>
<td>G-2</td>
<td>E-2</td>
<td>F-2</td>
<td>E-1</td>
</tr>
<tr>
<td>G-2</td>
<td>F-2</td>
<td>E-2</td>
<td>F-2</td>
<td>E-1</td>
</tr>
<tr>
<td>Norm</td>
<td>482</td>
<td>332</td>
<td>370</td>
<td>201</td>
</tr>
<tr>
<td>EC</td>
<td>0</td>
<td>8</td>
<td>29</td>
<td>21</td>
</tr>
</tbody>
</table>
CHAPTER V

EVALUATION OF TECHNIQUES

In order to compare and evaluate the techniques for automatic seriation which we described in Chapter II, we constructed two systems of computer programs. The first (PHOENIX-I) was programmed at Rice University in 1965; this was partially rewritten and extended as PHOENIX-II at Carnegie-Mellon University in 1966. A user's guide to PHOENIX-II is included here as Appendix C; the actual program listing is provided in Appendix D; and Appendix E contains some comments on the organization of the program and changes necessary to convert it to another computer.

As the two versions of PHOENIX are conceptually very much alike, we will not attempt to distinguish between them in this discussion. We will refer instead to an abstract system PHOENIX, which contains both PHOENIX-I and -II as well as some features which have been designed but not yet implemented.

PHOENIX provides a capability for reading a set of data (raw counts, tabulation of percentages, precomputed correlation matrices) and applying to that set of data a selection of programs which carry out the various ordering techniques that we wish to use or evaluate. After the seriation procedures are carried out, we can obtain output in a variety of forms: printouts of raw data, percentages, correlation matrices, contour plots, and type-percentage plots. With these capabilities, PHOENIX can do virtually any task demanded in seriation.

The precise mechanism for specifying (1) the nature of the data, (2) the ordering procedures to use and the order in which to apply them, and (3) the forms of output to be obtained for each case is more a part of the implementation of PHOENIX than it is of the abstract definition. The particular mechanism we use is described in Appendix C (the user's guide).

The Brainerd-Robinson Approach

The approach proposed by Brainerd and Robinson was not the first attempt (see Brainerd 1951:304, fn.5) to set up formal criteria for deducing the chronological order inherent in anthropological data, but it stands today as the most used and, we believe, the most successful of the various proposals. The rationale they use for defining the ideal form of an ordered matrix has not been contradicted; indeed, it has served as the basis for a
number of variations on the original proposal. For example, their ordering criteria apply even when the correlation coefficients are computed somewhat differently, as in the Dempsey-Baumhoff technique, and when different norms are used for deciding on the “best” order for the data (see Chapter III and Kuzara, Mead and Dixon 1966). We have also found it helpful to follow Brainerd’s suggestions for using contour maps of the correlation matrices as an aid to interpreting data. Permutation Search, which allows the archeologist to influence the progress of the seriation after checking preliminary output, is particularly able to take advantage of the graphic nature of the contour displays.

There were a number of questions left open by the original Brainerd and Robinson papers, the chief one of which was how one actually goes about discovering the order of sites which gives the best approximation to the “ideal” form. Most of the effort expended on the Brainerd-Robinson approach has been directed at this problem. A related problem is finding a suitable norm (or “matrix coefficient”) for evaluating a given order. We discussed this in Chapter III.

A third problem related to finding the best order is the question of when to stop searching. This problem crops up chiefly in manual searching, where it is complicated by the subjectivity of the human orderer. An example may be found in Troike (1957). For La Venta pottery he finds A B C D E 1 2 3 4 5 to be a good order for the components, except for A, in which he suspects mixing. He stops searching at this point, having confirmed the excavator’s stratigraphy and separation of trenches. Permutation Search, however, shows that the order D B C A E 1 2 3 4 5 is substantially better, having a norm of seven as opposed to a norm of 119 for the stratigraphic order.

Part of the problem in this case relates to whether we wish to find the best order inherent in the data or the “best” order in archeological fact. In Troike’s words, “virtually all of Drucker’s conclusions are confirmed through the application of the Brainerd-Robinson seriation technique, a point of no small importance to archeological theory” (Troike 1957:282). Unfortunately, this is not true. Permutation Search shows the stratigraphic order in one trench to be good and that the two trenches are separate in time, but it also indicates that there is considerable mixing of the deposits or some other kind of difficulty with the data.

**The Ascher Technique**

We obtained a program listing and prose description of the program developed by Ascher and Ascher (1963) and edited it slightly to run as a part of the PHOENIX system; we are, however, satisfied that our version is substantially equivalent to that of the Aschers.

Although the program does well with data which are very well condi-
tioned (nearly ideal), our experience in using it on other data of known order has been highly unsatisfactory (see Tables 4-2 to 4-9). The program is more sensitive to the input order than any of the other techniques we have examined. Indeed, the order produced by the program is frequently worse (i.e., has a higher norm) than the order in which the data were submitted. Kuzara, Mead and Dixon (1966) reported similar results with the Ascher program.

For several of our data cases, we cycled repeatedly through the Ascher program, using the order found by one trial as the input order for the next. In a large number of instances we found an erratic sequence of orderings, jumping from fairly good to very bad and back without any evidence of a trend toward overall improvement. This chain of unrelated orderings in some cases terminated when one order (not usually the best) became stable; that is, when given this order to start, the Ascher program produced the same order as a result. The rest of the cases terminated in a repeating cycle of orderings. For example, given some input order, the Ascher program produced a second; given the second, it produced a third; given the third, it produced the first; and so forth until we stopped the run.

The major difficulty with the Ascher program is the way it constructs the matrix row by row, instead of considering it as a whole. This is illustrated by the examples that follow (see Fig. 5-1).

The matrix in Figure 5-1(a) has a norm of 1 (in the order given) and is, at least intuitively, in the best order to fit the Brainerd-Robinson model. Now assume that the Ascher program begins with the order given in Figure 5-1(b). The first step is to attempt to place site C; this is done in Figure 5-1(c). The algorithm gives only one alternative, and there is no question that it applies—site C must lie between sites A and D. Unfortunately, this is not consistent with the best order given in Figure 5-1(a). The misplacement was permitted because the entries corresponding to sites B and E were completely ignored. The ordering then proceeds as in Figures 5-1(d) and 5-1(e). In Figure 5-1(e) not only are all of the possibilities deferred by the first phase of the Ascher procedure, but all of these possible orderings have norms greater than 1. Thus, even the Aschers’ second phase cannot find the best order because sites cannot be moved once they are placed.

The Ascher program makes no provisions for detecting such situations and the user, especially if he is not a computer programmer, does not have the control over the ordering procedure which might allow him to overcome these problems.

The Meighan Technique

Although the Meighan technique is conceptually a part of the PHOENIX system, it has not been implemented as a program in either of the two
Figure 5-1. Ordering a correlation matrix by the Ascher method (data hypothetical).

versions now in operation. There are two reasons for this: first, the 3-pole plots are not difficult to generate by hand; second, we feel that the method can be extended to handle an arbitrary number of types and we prefer to wait and program the more general form of the technique.

The hand-drawn 3-pole plots are not completely accurate, but they do provide a fairly good estimate of an order for the data. We have found this to be a useful preordering technique for use in conjunction with Permutation Search; we find gains both in the ultimate accuracy and in the
computation time. By providing approximate placements for the sites, a 3-pole preordering may prevent Permutation Search from getting stuck in an apparent best fit which has gross reversals in the sequence (e.g., see p. 21). In addition, the overall ordering, even if rough, may get the sequence well enough sorted to save one or two iterations with Permutation Search. This is especially likely to happen for large cases, where the savings in machine time are greatest.

Potentially one of the most valuable uses of the 3-pole approach is in the preliminary evaluation of a data set (see discussion on p. 37). The major problem in this is the selection of the types to plot. As we discussed with regard to the characteristics of archeological data, this judgment must be based on specific information for each culture area.

To make a fair comparison between the operation of the 3-pole technique and that of Permutation Search, we followed Meighan’s suggestion and selected the three most numerous types in each set. From Data Set I, we chose two types of Susiana Black-on-Buff and one of Susiana Buff (Table 5-1). The resultant graph (Fig. 5-2) shows that components A-3, A-2, and A-1 split off widely to one side, so we considered the data in separate sets and drew two lines through the points. By this method, A-3, A-2, and A-1 seriated in the correct order but there was no way to relate that set to the remaining components or even to tell whether A-3 or A-1 should be adjacent to either M-E or D, the end points on the other axis. Reference to Table 5-1 shows that the 3-pole results duplicated the order obtained by Permutation Search, except for the difficulty of relating A-1, A-2, and A-3 to the rest of the components.

Since the distribution of Susiana Buff was thought to vary with the distribution of Susiana Black-on-Buff (see p. 56), another 3-pole graph based

---

**TABLE 5-1**

DATA FOR 3-POLE PLOT OF DEH LURAN POTTERY.

<table>
<thead>
<tr>
<th>Types Deh Luran Pottery</th>
<th>Data Set I</th>
<th>Resultant Order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>A-1</td>
<td>55</td>
<td>48</td>
</tr>
<tr>
<td>A-2</td>
<td>101</td>
<td>88</td>
</tr>
<tr>
<td>A-3</td>
<td>127</td>
<td>150</td>
</tr>
<tr>
<td>M-E</td>
<td>20</td>
<td>13</td>
</tr>
<tr>
<td>B-1</td>
<td>132</td>
<td>147</td>
</tr>
<tr>
<td>B-2</td>
<td>156</td>
<td>131</td>
</tr>
<tr>
<td>B-3</td>
<td>81</td>
<td>80</td>
</tr>
<tr>
<td>C-1</td>
<td>40</td>
<td>78</td>
</tr>
<tr>
<td>C-2</td>
<td>8</td>
<td>45</td>
</tr>
<tr>
<td>C-3</td>
<td>4</td>
<td>42</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>235</td>
</tr>
</tbody>
</table>

EC 6 6
Figure 5-2. 3-pole plot of types in Table 5-1.

on the three most numerous Black-on-Buff types (Fig. 5-3, Table 5-2) was set up. Again the results required lines fitted to two subsets of the data because the early end split off from the remainder. The order was quite good, having an Error Coefficient of 4. The Permutation Search result was also affected by this break between the two sets of components. The data set with only three types did not provide enough information to locate these components properly.

The splitting off of three components on each plot was caused by a great change in percentage of one of the types in each set. In the first

<table>
<thead>
<tr>
<th>TABLE 5-2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATA FOR 3-POLE PLOT OF DEH LURAN POTTERY, TYPES 15, 23, 25 (SUSIANA BLACK-ON-BUFF)</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>15</th>
<th></th>
<th>23</th>
<th></th>
<th>25</th>
<th></th>
<th>( \Sigma )</th>
<th></th>
<th>Resultant Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>%</td>
<td>N</td>
<td>%</td>
<td>N</td>
<td>%</td>
<td>N</td>
<td>%</td>
<td>3-Pole</td>
<td>PS</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>A-1</td>
<td>-</td>
<td>55</td>
<td>53.4</td>
<td>48</td>
<td>46.6</td>
<td>103</td>
<td>A-3</td>
<td>C-2</td>
<td></td>
</tr>
<tr>
<td>A-2</td>
<td>1</td>
<td>0.5</td>
<td>101</td>
<td>53.2</td>
<td>88</td>
<td>46.3</td>
<td>190</td>
<td>A-1</td>
<td>C-1</td>
</tr>
<tr>
<td>A-3</td>
<td>6</td>
<td>2.1</td>
<td>127</td>
<td>44.9</td>
<td>150</td>
<td>53.0</td>
<td>283</td>
<td>A-2</td>
<td>B-3</td>
</tr>
<tr>
<td>M-E</td>
<td>4</td>
<td>10.8</td>
<td>20</td>
<td>54.1</td>
<td>13</td>
<td>35.1</td>
<td>37</td>
<td>B-1</td>
<td>M-E</td>
</tr>
<tr>
<td>B-1</td>
<td>12</td>
<td>4.1</td>
<td>132</td>
<td>45.4</td>
<td>147</td>
<td>50.5</td>
<td>291</td>
<td>M-E</td>
<td>B-2</td>
</tr>
<tr>
<td>B-2</td>
<td>23</td>
<td>7.4</td>
<td>156</td>
<td>50.3</td>
<td>131</td>
<td>42.3</td>
<td>310</td>
<td>B-3</td>
<td>B-1</td>
</tr>
<tr>
<td>B-3</td>
<td>55</td>
<td>25.5</td>
<td>81</td>
<td>37.5</td>
<td>80</td>
<td>37.0</td>
<td>216</td>
<td>B-2</td>
<td>A-3</td>
</tr>
<tr>
<td>C-1</td>
<td>105</td>
<td>47.1</td>
<td>40</td>
<td>17.9</td>
<td>78</td>
<td>35.0</td>
<td>223</td>
<td>C-1</td>
<td>A-2</td>
</tr>
<tr>
<td>C-2</td>
<td>53</td>
<td>50.0</td>
<td>8</td>
<td>7.5</td>
<td>45</td>
<td>42.5</td>
<td>106</td>
<td>C-2</td>
<td>A-1</td>
</tr>
<tr>
<td>C-3</td>
<td>10</td>
<td>17.9</td>
<td>4</td>
<td>7.1</td>
<td>42</td>
<td>75.0</td>
<td>56</td>
<td>C-3</td>
<td>C-3</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>1</td>
<td>0.4</td>
<td>235</td>
<td>99.6</td>
<td>235</td>
<td>D</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

EC 4 7
case, type 42 of the Susiana Buff jumped from 13 to 57 per cent between components A-3 and B-1. In the second case, two types each changed about 20 per cent between components B-2 and B-3. In neither of these examples do we have a "continuous" series in the ideal sense, if we take the evidence of only three types.

Neither of the results was an improvement over the best orderings obtained from a coefficient matrix based on Data Set I, but each gave a close approximation to the true order and, in view of the ease with which 3-pole graphs can be constructed, it seems especially appropriate to use them for preliminary analysis in the field or for preordering a matrix for other programmed techniques.

Another interesting effect occurred when we tried a 3-pole ordering of the unabbreviated Pa Sangar flints, again using the three most numerous types (Fig. 5-4, Table 5-3). The plot gave an order which is roughly consistent with the stratigraphy although it has an Error Coefficient of 42. A comparison between the 3-pole ordering and the Permutation Search result, using the same three types as a data set, is revealing. The Meighan technique uses a straight line to approximate the chronological order; Permutation Search seriated most of the sites in sequences which involved moving the shortest distances across the 3-pole graph at each step (Fig. 5-5). This indicates that the two techniques do not measure the same things. In terms of the stratigraphy, the Permutation Search ordering is somewhat better, but both seriations are fairly good, having lower Error Coefficients.
### TABLE 5-3

**Data for 3-Pole Plot of Pa Sangar Flint, Types 18, 26, 36**

<table>
<thead>
<tr>
<th></th>
<th>18</th>
<th></th>
<th>26</th>
<th></th>
<th>36</th>
<th></th>
<th>Σ</th>
<th>Resultant Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>23</td>
<td>62</td>
<td>1</td>
<td>6</td>
<td>30</td>
<td>34</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>52</td>
<td>85.2</td>
<td>7</td>
<td>11.5</td>
<td>3.3</td>
<td>3.3</td>
<td>61</td>
<td>12, 12</td>
</tr>
<tr>
<td>23</td>
<td>74</td>
<td>83.1</td>
<td>11</td>
<td>12.4</td>
<td>4</td>
<td>4.5</td>
<td>89</td>
<td>23, 23</td>
</tr>
<tr>
<td>34</td>
<td>40</td>
<td>85.1</td>
<td>7</td>
<td>14.9</td>
<td></td>
<td></td>
<td>47</td>
<td>56, 45</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>69.6</td>
<td>19</td>
<td>27.5</td>
<td>2</td>
<td>2.9</td>
<td>69</td>
<td>01, 01</td>
</tr>
<tr>
<td>56</td>
<td>153</td>
<td>79.7</td>
<td>26</td>
<td>13.5</td>
<td>13</td>
<td>6.8</td>
<td>192</td>
<td>78, 67</td>
</tr>
<tr>
<td>67</td>
<td>129</td>
<td>75.9</td>
<td>34</td>
<td>20.0</td>
<td>7</td>
<td>4.1</td>
<td>170</td>
<td>67, 56</td>
</tr>
<tr>
<td>78</td>
<td>131</td>
<td>78.9</td>
<td>21</td>
<td>12.7</td>
<td>14</td>
<td>8.4</td>
<td>166</td>
<td>45, 78</td>
</tr>
<tr>
<td>89</td>
<td>65</td>
<td>71.4</td>
<td>10</td>
<td>11.0</td>
<td>16</td>
<td>17.6</td>
<td>91</td>
<td>89, 89</td>
</tr>
<tr>
<td>90</td>
<td>35</td>
<td>67.3</td>
<td>7</td>
<td>13.5</td>
<td>10</td>
<td>19.2</td>
<td>52</td>
<td>90, 90</td>
</tr>
<tr>
<td>101</td>
<td>57</td>
<td>61.3</td>
<td>18</td>
<td>19.4</td>
<td>18</td>
<td>19.4</td>
<td>93</td>
<td>101, 101</td>
</tr>
<tr>
<td>112</td>
<td>46</td>
<td>39.7</td>
<td>8</td>
<td>10.4</td>
<td>23</td>
<td>29.9</td>
<td>77</td>
<td>123, 123</td>
</tr>
<tr>
<td>123</td>
<td>31</td>
<td>38.5</td>
<td>12</td>
<td>22.6</td>
<td>10</td>
<td>18.9</td>
<td>53</td>
<td>189, 190</td>
</tr>
<tr>
<td>134</td>
<td>34</td>
<td>39.6</td>
<td>8</td>
<td>14.0</td>
<td>15</td>
<td>26.3</td>
<td>57</td>
<td>134, 112</td>
</tr>
<tr>
<td>145</td>
<td>80</td>
<td>60.2</td>
<td>14</td>
<td>10.5</td>
<td>39</td>
<td>29.3</td>
<td>133</td>
<td>145, 145</td>
</tr>
<tr>
<td>156</td>
<td>39</td>
<td>56.5</td>
<td>11</td>
<td>15.9</td>
<td>19</td>
<td>27.5</td>
<td>69</td>
<td>112, 134</td>
</tr>
<tr>
<td>167</td>
<td>18</td>
<td>56.2</td>
<td>5</td>
<td>15.6</td>
<td>9</td>
<td>28.1</td>
<td>32</td>
<td>156, 189</td>
</tr>
<tr>
<td>178</td>
<td>17</td>
<td>45.9</td>
<td>8</td>
<td>21.6</td>
<td>12</td>
<td>32.4</td>
<td>37</td>
<td>190, 156</td>
</tr>
<tr>
<td>189</td>
<td>28</td>
<td>39.6</td>
<td>7</td>
<td>14.9</td>
<td>12</td>
<td>25.5</td>
<td>47</td>
<td>167, 167</td>
</tr>
<tr>
<td>190</td>
<td>13</td>
<td>59.1</td>
<td>2</td>
<td>9.1</td>
<td>7</td>
<td>31.8</td>
<td>22</td>
<td>201, 201</td>
</tr>
<tr>
<td>201</td>
<td>9</td>
<td>52.9</td>
<td>4</td>
<td>23.5</td>
<td>4</td>
<td>23.5</td>
<td>17</td>
<td>EC 42, 178</td>
</tr>
</tbody>
</table>

Figure 5-4. 3-pole plot of types in Table 5-3.
The label on each point is the first two digits of the three-digit component label.
than any of the orderings of the complete data set except Permutation Search on the Brainerd-Robinson matrix. This example gives firm support to the use of some approach like the 3-pole technique for preordering data to be seriated.

When the seriations from 3-pole orderings were used as input for Permutation Search, the results were frequently considerably better than Permutation Search alone or any other method or combination of methods had been able to achieve. It should also be noted here that the "best" order in terms of the matrix norm was in some cases not very close to the true stratigraphic sequence.

One reason that we find relatively good initial approximations with the 3-pole approach is that our data are dominated by few types and, as a result, the three types selected for the approximation tend to account for the bulk of the total data set. Meighan anticipated this when he remarked that his method "yields results that are often as accurate as those methods using the full range of types. The reason for this is that most collections have a small number of components as the bulk of the collection, other items being numerically, and hence statistically, an insignificant portion of the total" (1959:204-205).

Meighan noted (1959:204) that the best fit might often be a curved line, but that mathematical techniques for finding the best curved fit are not available. The plots for Data Sets I and Ia bear this out, and suggest a partial solution. In order to obtain an ordering, we split each of these
data sets into two groups, found a best line for each group, and worried later about which way to join the two lines. In both cases, the proper result in terms of the ultimate best fit could have been obtained by extending the two lines until they crossed and fitting a curve which approximated the path of the joined lines. This suggests that a similar technique may be generally useful for data sets which show curved or discontinuous behavior. The curve of the line of best fit also bears on the validity of the data. In this regard we now present an argument to show that good data cannot possibly have a 3-pole graph which results in a closed curve, and derive from that argument a test which will tell when data are definitely not following a lenticular distribution.

Assume that the 3-pole plot of a data set is a closed curve. We can reconstruct the behavior of the types if we recall that for each type, the percentage is represented on the graph by the distance from the proper corner of the triangle to the curve—the closer the plot is to the point, the higher the percentage. Therefore, if the data form a closed curve, there must be a segment of the curve for each type (with an associated point of the 3-pole triangle) which bows away from the point of the triangle and then back toward it. If this were not true, the curve could not be closed. Each place where this behavior occurs is a situation where the data behave in a non-lenticular fashion—going from a greater to a lesser and then back to a greater percentage of a single type. A single instance of this type of behavior can be eliminated by allowing the starting and finishing point of the axis to fall between the ends of the bowed segment of the curve. For a closed curve, however, there will be three separate instances of this non-lenticular behavior, one for each of the three types. In such a case there will be at least one type for which the nonlenticular behavior remains in the ordering, no matter where the starting point is chosen. Thus, if a closed curve is obtained from a 3-pole ordering, at least one of the types involved violates our assumptions about data. It follows that good data cannot loop, and further that they cannot form a curve which is so sharply curved that it is not possible to distinguish its end points.

We now know that good data (Figs. 3-3, 3-4) and bad data can both result in curved 3-pole plots. From the analysis of looping behavior we can obtain a simple diagnostic test for distinguishing between these cases: Regard the 3-pole plot as a cross section of a surface, consider the three possible orientations which would result from placing each of the three sides of the triangle in a horizontal position, and ask whether water falling vertically onto the surface in such an orientation would run off or collect in a puddle. When a segment of a 3-pole plot bows away from some point of the triangle and back toward it, we shall then say that that segment of the curve "holds water." The diagnostic test consists simply of considering each point of the triangle in turn and checking to see whether any segment of the curve
“holds water” with respect to that point. If any segment of the curve holds water with respect to any of the points, then the corresponding type does not have a lenticular distribution.

The Dempsey-Baumhoff Technique

Dempsey and Baumhoff (1963:496) contend that contextual analysis yields results which compare favorably with results obtained by other methods, and it is relatively easy to apply. Setting aside for a moment any consideration of the merits of contextual analysis as a method, we take exception to the claim that it is easy to apply. As the original paper on the method amply documents, hand-computed matrices are subject to errors. In fact, the numerous computational errors in Dempsey and Baumhoff’s published matrices render their results invalid. We point this out, not as personal criticism, but rather as further evidence that suggests the importance of using purely mechanical, accurate, and unbiased machines for such complicated problems as computing and ordering matrices. By way of comparison, the result we obtained from programming the Dempsey-Baumhoff method and applying it to the Belous data, are contained in Table 5-4.

### TABLE 5-4

<table>
<thead>
<tr>
<th>Dempsey-Baumhoff Order</th>
<th>Computed Orders A</th>
<th>Computed Orders B</th>
<th>Averaged to Resolve Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>C66</td>
<td>S60-C</td>
<td>S60-C</td>
<td>S60-C</td>
</tr>
<tr>
<td>C141</td>
<td>C66</td>
<td>C66</td>
<td>C66</td>
</tr>
<tr>
<td>C107-AB</td>
<td>S99</td>
<td>C107-AB</td>
<td>C107-AB</td>
</tr>
<tr>
<td>S99</td>
<td>S99</td>
<td>S99</td>
<td>S99</td>
</tr>
<tr>
<td>S60-C</td>
<td>C107-C</td>
<td>C107-AB</td>
<td>C107-AB</td>
</tr>
<tr>
<td>C142-A</td>
<td>C142-A</td>
<td>C107-AB</td>
<td>C107-AB</td>
</tr>
<tr>
<td>S250</td>
<td>C142-B</td>
<td>C142-AB</td>
<td>C142-B</td>
</tr>
<tr>
<td>S66</td>
<td>S250</td>
<td>C142-B</td>
<td>C142-B</td>
</tr>
<tr>
<td>C150</td>
<td>S56</td>
<td>S66</td>
<td>S66</td>
</tr>
<tr>
<td>S1-A</td>
<td>C107-AB</td>
<td>C107-C</td>
<td>C107-C</td>
</tr>
<tr>
<td>C107-C</td>
<td>S66</td>
<td>C56</td>
<td>S66</td>
</tr>
<tr>
<td>C142-B</td>
<td>C68</td>
<td>S250</td>
<td>S250</td>
</tr>
<tr>
<td>S60-AB</td>
<td>C150</td>
<td>S1-B</td>
<td>S1-B</td>
</tr>
<tr>
<td>C56</td>
<td>S1-B</td>
<td>C150</td>
<td>C150</td>
</tr>
<tr>
<td>S1-B</td>
<td>S60-AB</td>
<td>S1-A</td>
<td>S1-A</td>
</tr>
<tr>
<td>C68</td>
<td>S1-A</td>
<td>S60-AB</td>
<td>S60-AB</td>
</tr>
</tbody>
</table>

(1) We attained the oscillation anticipated by Dempsey and Baumhoff (1963:504); computed orders A and B give the two sets of results. We resolved the impasse in the final column by following their suggestions about averaging.

(2) If we assume our average order is correct, the Dempsey-Baumhoff order has an Error Coefficient of 47; if we assume their order is correct, the averaged order has an Error Coefficient of 57. Note that neither of these Error Coefficients says anything about relations to true order—which we don’t know—but they do give an order of the magnitude of error produced by small arithmetic errors.

We distinguish between two proposals made by Dempsey and Baumhoff: first, the use of a presence-absence matrix to express the similarity among sites; and second, the use of Contextual Analysis to obtain the orderings. For our purposes this division is justified because the presence-absence
matrices have the same ideal form and behave in the same manner as Brainerd-Robinson matrices.

**Presence-Absence Coefficients:** We have already commented on some of the effects of using presence-absence coefficients in the correlation matrix. On the whole, we have found that Brainerd-Robinson coefficients produce the best results. The only case where we obtained better results with presence-absence coefficients was with the reduced form of Data Set III (Table 4-8b). The Pa Sangar data were not particularly well conditioned and the Presence-Absence matrix proved more sensitive to nonlenticular variation.

It seems fair to interject here at least three possible sources of error to which a Presence-Absence matrix is subject. First, it is sensitive to mixing of deposits. A stray artifact out of stratigraphic place can introduce a disproportionate amount of noise into the analysis. For this reason, the strength of the Presence-Absence matrix over the Brainerd-Robinson form is mitigated and perhaps obviated for many sites. Secondly, types whose attributes vary along a continuous metric dimension may change gradually from one modal type to another. In these cases, it is likely that the chronologically important characteristics can only be expressed as variations in frequency. Presence-Absence matrix will only record the presence of all types and hence not be useful for discriminating temporal differences (see Lipe [1964] for further discussion of these points). A final problem concerns the use of assemblages of widely varied artifacts. Lack of weighting can be a virtue but it seems, at least intuitively, that including "types" of burials, "types" of pottery, and "types" of houses in one matrix raises serious problems.

**Contextual Analysis:** The program which was written to handle Contextual Analysis performs all the operations for which the computational methods are specified in Dempsey and Baumhoff's original description. That is, the program carries out the method as described through the iterated step: it splits the sites into two groups; computes the average agreement of each site with members of the other group; reorders on the basis of the new average agreements, and tries again until no regrouping takes place. Our program does not attempt to eliminate "Pattern III" artifacts.

In fact, we firmly disagree with the attempt to eliminate these artifacts. The Brainerd-Robinson matrix is designed to take them into account with no particular trouble and since they will occur in any long sequence, we feel that they should be allowed to contribute to a seriation. Further, Pattern III artifacts should not interfere significantly with a seriation because the distribution of other types should indicate where the sites which contain these artifacts belong.

Sets of perfect data are more difficult to obtain than are sets that contain
types suitable for seriation. We prefer to deal with the types on their own terms rather than judge them on the basis of their occurrence in a particular matrix. There is also the possibility that after removing Pattern III artifacts and reseriating the matrix, the new order may contain new Pattern III artifacts while some of the old Pattern III artifacts would now fit.

With our program for Contextual Analysis as it stands, we are able to obtain orderings which are generally correct but have errors in some details. The Contextual Analysis orderings, like 3-pole seriations, are most useful when taken as preliminary orderings for Permutation Search.

Contextual Analysis, like the 3-pole technique, has the advantage that it is independent of input order. It is iterative, however, and therefore somewhat slower than a programmed 3-pole ordering would be. Moreover, Contextual Analysis does not work directly toward our stated goal (minimization of a norm over a correlation matrix) but rather toward a goal of its own. As a result, an ordering which we consider "worse" may supersede a "better" order as in Data Set 1a. This is not an inherent flaw, but it sometimes works against the Brainerd-Robinson goal when Contextual Analysis is used to find a preordering, and we do not find the technique strong enough to stand alone.

We do feel that we have lost none of the strength of Contextual Analysis by omitting the elimination of suspected "Pattern III" artifacts. As described above, we attempted to seriate Data Set I using the Dempsey-Baumhoff method on a Brainerd-Robinson matrix. A straight run with no attempt made to compensate for "difficult types" produced an order with a norm of 547. In general terms this ordering is not bad since there is no extreme mixing, but it is far from being as good as the best order obtained on the same matrix with Permutation Search.

If we follow the Dempsey-Baumhoff method strictly, after a trial ordering we must eliminate the types that do not occur in one of the end sites. Dempsey and Baumhoff acknowledge that while all their types have a curvilinear relationship to time, some may not be represented in all periods during a span of archeologic time. The crucial factor in eliminating "Pattern III" artifacts in Dempsey and Baumhoff's method is the distribution of types within the span under analysis and not the behavior of the types per se.

To obtain an evaluation of the technique independent of the Pattern III question we set up two data sets which we knew to contain only artifacts of Patterns I, II, and IV. The results of running subsets of Data Set I with Pattern III artifacts removed are summarized in Tables 5-5 and 5-6. Only one of the results (Permutation Search on a Brainerd-Robinson matrix with Data Subset 1b) was as good as obtained by running the full set using Permutation Search on a Brainerd-Robinson matrix. In each case, using the Dempsey-Baumhoff method, the results were inconsistent with the actual stratigraphy. The most serious error however, was in the placing of
components in incorrect phase contexts. In other words, the A’s, B’s, and C’s should have been grouped. Still, all of the orders except Contextual Analysis on a Brainerd-Robinson matrix were reasonably close approximations to the true order and very acceptable as preorderings.

**The Permutation Search Technique**

There were several separate tasks involved in the development of Permutation Search. The first was to find a seriation method which worked toward the goal stated by Brainerd and Robinson. The second task was a theoretical problem: to find a flexible technique that could be conveniently modified at a number of points. The most important of the points

---

**TABLE 5-5**

**DATA SUBSET Ia, PATTERN III ARTIFACTS REMOVED**

**SUSIANA BLACK-ON-BUFF, DEH LURAN POTTERY**

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>DB</th>
<th>PS</th>
<th>DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-1</td>
<td>A-3</td>
<td>B-1</td>
</tr>
<tr>
<td>A-2</td>
<td>A-3</td>
<td>A-2</td>
<td>A-2</td>
</tr>
<tr>
<td>A-3</td>
<td>B-1</td>
<td>A-1</td>
<td>A-3</td>
</tr>
<tr>
<td>ME</td>
<td>A-2</td>
<td>ME</td>
<td>A-1</td>
</tr>
<tr>
<td>B-1</td>
<td>ME</td>
<td>B-2</td>
<td>B-2</td>
</tr>
<tr>
<td>B-2</td>
<td>B-2</td>
<td>B-3</td>
<td>B-3</td>
</tr>
<tr>
<td>B-3</td>
<td>B-3</td>
<td>B-1</td>
<td>ME</td>
</tr>
<tr>
<td>C-1</td>
<td>C-1</td>
<td>C-3</td>
<td>C-3</td>
</tr>
<tr>
<td>C-2</td>
<td>C-3</td>
<td>C-1</td>
<td>C-1</td>
</tr>
<tr>
<td>C-3</td>
<td>C-2</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>E-1</td>
<td>E-2</td>
<td>E-2</td>
<td>E-2</td>
</tr>
<tr>
<td>E-2</td>
<td>E-1</td>
<td>E-1</td>
<td>E-1</td>
</tr>
</tbody>
</table>

Norm\(^*\) 48 344 1253
EC 7 11 13

\*Norms not comparable

**TABLE 5-6**

**DATA SUBSET Ib, PATTERN III ARTIFACTS REMOVED**

**SUSIANA BUFF, DEH LURAN POTTERY**

<table>
<thead>
<tr>
<th>Initial Order</th>
<th>DB</th>
<th>PS</th>
<th>DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>A-1</td>
<td>A-3</td>
<td>A-3</td>
</tr>
<tr>
<td>A-2</td>
<td>A-3</td>
<td>A-1</td>
<td>A-1</td>
</tr>
<tr>
<td>A-3</td>
<td>ME</td>
<td>A-2</td>
<td>A-2</td>
</tr>
<tr>
<td>ME</td>
<td>A-2</td>
<td>ME</td>
<td>C-2</td>
</tr>
<tr>
<td>B-1</td>
<td>B-1</td>
<td>B-1</td>
<td>D</td>
</tr>
<tr>
<td>B-2</td>
<td>B-2</td>
<td>B-2</td>
<td>C-3</td>
</tr>
<tr>
<td>B-3</td>
<td>C-1</td>
<td>B-3</td>
<td>B-1</td>
</tr>
<tr>
<td>C-1</td>
<td>B-3</td>
<td>C-1</td>
<td>ME</td>
</tr>
<tr>
<td>C-2</td>
<td>C-3</td>
<td>C-2</td>
<td>B-3</td>
</tr>
<tr>
<td>C-3</td>
<td>C-2</td>
<td>C-3</td>
<td>C-1</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>D</td>
<td>B-2</td>
</tr>
</tbody>
</table>

Norm\(^*\) 60 204 1034
EC 6 2 13

\*Norms not comparable
of flexibility are computation of matrix entries, norm, and search pattern. A third task was to set up a program with the desired flexibility. This became the Permutation Search technique, with correlation matrix, norm, and search pattern provided by the user (but not necessarily as a part of every data case) and a choice between two improvement techniques. Fourth, we selected two forms of correlation matrix, a norm, and two search patterns for intensive study and wrote programs to perform the appropriate calculations. Fifth, we postulated and verified a need for a conversational system, with the user adding his knowledge and the results of other programs to the resources of Permutation Search. The sixth was to provide a capacity for constraining the order of subsets of the sites in order to take advantage of prior knowledge of chronology and to reduce computation time. The Type-Percentage variation of the standard Permutation Search was the seventh proposal.

The original decision to work with all the data and stick closely to the Brainerd-Robinson criteria was, we believe, a good one. The results produced by our programs for Permutation Search are consistently more successful than the other techniques we have examined and most of the shortcomings could probably be at least partially overcome by using other search patterns. The dependence on starting order, although it could be reduced in severity by developing other search patterns, is an inherent problem with a searching, trial-and-error technique of this type. The tendency to leave entire sets of sites misplaced overall but well ordered within the sets could, we believe, be solved by setting up a search pattern which concentrates on gross aspects of the ordering as well as (or in addition to) the finer detail to which we presently devote our attention. We do not feel pressed to develop such patterns, however, because the gross displacements are easy to spot visually and fix.

One of the major strengths of Permutation Search is its flexibility. It is a simple task to change the computation of coefficients in the correlation matrix, the norm, the search patterns, or even the schemes for immediate and delayed improvement. It is conceptually simple to alter the abstract description of one of these computations; in PHOENIX-II (see Appendix D) each change involves only the substitution of a well-isolated subprogram varying from a third of a page to slightly more than a page in length. The flexibility of the system allows us to test our own theories, and gives us the ability to take advantage of programs developed by other persons—an important factor in a field as active as automatic seriation is at present. A consideration which is secondary to us but may be important to those who wish to use PHOENIX-II is that we have implemented it in such a way that others should find it easy to substitute subprograms to test their own ideas or to fit special aspects of their own data.

We have discussed above (sections on Brainerd-Robinson technique, p.
8, Dempsey-Baumhoff technique, p. 11, and Permutation Search, p. 13) our choices of methods for computing the correlation matrix; we have found that the Brainerd-Robinson method usually extracts more useful information from the data than the presence-absence form. Also discussed above (p. 28) are the reasons for our choice of a simple sum of errors for the matrix norm. This is the only one we have used seriously. We take the position that good data should give identical or almost identical orderings under all reasonable norms, and that variability of results produced by switching norms probably indicates poor data.

Both of our search procedures were developed early in our work with automatic seriation. One is, as noted above, almost identical with the only technique provided by the Kuzara, Mead, and Dixon program; the other was developed on similar intuitive grounds. We have, on the whole, been well satisfied with our search patterns. We see the immediate and delayed improvement schemes as two ways to take advantage of information obtained through permutation searching. At the moment, we have no grounds for choosing between them.

As we worked with seriation techniques and real data sets, we found that routine and invariable tabulation of the contents of sites did not convey to the programs all of the information we might have about the sequence within which we were trying to deduce chronology. Independent information on order could and should be expressed concisely as constraints on the permissible resultant orders (this has not yet been programmed); other information was qualitative (such as interpreting contours) and could only be incorporated by adding human judgment to the seriation system. As a consequence of these experiences, we feel at present that a better picture can be obtained by an informed human coordinating the results of several programs than by dependence on a single program. We feel that this type of organization is quite reasonable in view of the rapidly growing trend toward computer systems with a larger number of conversational consoles (or inquiry stations) set up at locations far from the computer. (We know of one such console located in Pittsburgh, Pennsylvania during the autumn of 1966 which allowed conversational calculation with a computer in Santa Monica, California.) Further, many of these interactive consoles are portable, even suitcase-sized. As a result, it seems reasonable to expect that within the next decade any worker with a serious need for computing power will be able to compute with his data, at his own convenience, and get tolerably fast response from the computer.

The Type-Percentage Technique

This is a variant of Permutation Search which came about as a result of working with the flexible system which has been described. As we looked into the relations among the numerous variable components it became clear
that techniques similar to those for correlation matrices could be applied to
the type-percentage tables to demand as close an approximation as possible
to the "ideal" lenticular form for all types. The only significant change in
PHOENIX-II necessary to accommodate this technique was the addition
of a program which computes, on request, a norm based on the "lenticu-
larity" of the type-percentage table rather than on the contours of a correla-
tion matrix. We have not yet completed the evaluation of this technique as
a tool in automatic seriation, but whatever its ultimate value, the pattern of
its development illustrates some of the strengths of Permuation Search in
the theoretical area.
CHAPTER VI
CONCLUDING REMARKS

We should emphasize at the start that seriation does work. It is a valid method for deriving chronology from archeological data, but it is not a method that can be used uncritically or universally. The remarks that follow are intended to reemphasize the potential shortcomings of seriation as a technique and to suggest some things that archeologists should consider when they attempt to order their data chronologically. It is safe to say that at this point the mechanical techniques of seriation are far more sophisticated than either the archeological data themselves or the theoretical understandings about the data and their cultural context. In this monograph we have identified many of the actual or potential problems and have suggested some of the ways they may be overcome. The remarks that deal with the formal and mechanical aspects of designing systems for seriation are intended as guidelines for other programmers. These remarks are based on our experience with the algorithms described in this study and on the designing of computer-based systems for other kinds of problems. We conclude the study with a list of open questions that suggest possible avenues of future research in the areas of seriation and of programming systems for seriation.

This study started with the premises that chronological ordering of archeological data is necessary for the reconstruction of culture history and that no single method of dating is universally applicable. It follows that alternative dependable techniques need to be developed and verified. This study has considered the technique of dating by seriation which is based on theoretical expectations of systematic changes in artifacts. In its general form, the proposition that artifacts change systematically with respect to time has never been severely challenged, but this study is one of the few attempts to test its application in particular cases where the true chronology is known beforehand.

We began by considering procedures that others had developed to accomplish the comparisons of sites that were necessary to deduce a chronological order from changes in artifacts. This in turn led us to undertake a testing of automatic or mechanical procedures to attempt to evaluate which were most efficient and accurate in terms of the theoretical goals of seriation.

Finally, since the work involved in such testing is enormous, we pro-
grammed the various proposals for a digital computer. In consequence of this, we eventually developed what we consider to be a very useful and flexible new system for carrying out the numerical operations incident to the analysis of archeological data and to displaying the results of this type of analysis.

The results of this study include conclusions, general guidelines, and recommendations, and open questions concerning seriation. For convenience in presentation the conclusions are divided into three parts: theoretical-cultural, archeological, and mechanical.

**Theoretical-Cultural Considerations**

When we set up data for seriation we tacitly assume a closed cultural system, whereas we actually know that virtually all cultural systems are somewhat open. The Brainerd-Robinson model of change is based on a system where the only extra-cultural variable is time. In this system changes are continuously engendered internally, growing out of local cultural circumstances.

Even the most casual view of cultures reveals that this model is too simple. Every culture is affected by external factors, many of which can influence culture change. One of the most obvious of these factors is migration into an area of people with new customs and strange artifacts, resulting in a discontinuity in the local sequence of types of artifacts. Similar discontinuity can arise from trade or from the borrowing and local adaptation of techniques of manufacture. Natural events such as climatic change, the exhaustion of soil, or extensive erosion may also induce disjunctive changes in the pattern of life and hence on the pattern of artifacts.

The simple model also assumes that the various parts of a contemporary cultural system are homogeneous, but we can cite many instances where this is not true. Specifically, if there are different kinds of sites (e.g., camps, villages, towns, markets), or much variety within one site (houses, temples, cemeteries), the differences among the artifacts found in each of these areas may be confused with chronological differences if the excavated samples are too small or if basic knowledge of the culture in question is too limited. Furthermore, if materials from too wide a geographic area are considered, differences that are not based on chronology may be interpreted as if they were.

We should also recognize that change, as we can see it in artifacts, proceeds at uneven rates. The common notion that cultural change is inevitable says nothing about its rate, and if change is minor during the span under investigation, it may not be possible to distinguish chronological differences even when other conditions are satisfied.

Archeologists infer cultural systems from artifacts and other remains, and they convert these data to a useful form for seriation by constructing a
typology of the artifacts which can be reduced further to a set of numbers. The kinds of artifact types that seriate best are well illustrated in Brainerd's paper, but finding such types archeologically is another matter. As we have shown, not all types seriate in all data sets according to the theoretical expectations. Within any cultural matrix, with respect to a typology of artifacts, the following may have to be considered.

The types may be "good," in which case they are commonly present in their appropriate period, and they are restricted—either by their characteristic frequency or by their presence and absence—to certain periods. Our studies suggest that the types most likely to satisfy these conditions are those which are made to conform to well-defined specifications (these can include tools, styles, designs) or which are the result of a characteristic manufacturing technique that is restricted to certain periods. For practical purposes these criteria suggest that good types are likely to have a relatively neutral adaptive value in a culture. Changes in them will derive more from fashion or from random drift than from necessity. We must not categorically exclude tools from consideration, however, for two reasons: first, changes in tools are in the long run a function of time, and second, in paleolithic sites tools are virtually the only kind of artifacts found. The crucial point is that only with respect to a particular problem can one determine which types are suitable. In the case of the paleolithic, we frequently deal with long periods of chronological time during which such artifacts change within relatively closed and homogeneous cultural systems.

"Bad" types can have a number of other attributes. They may be highly related to nontemporal activities, in which case their changes will be unpredictable. Types that are fortuitous (e.g., flakes that were picked up and used because they were handy for a job of the moment) are also not very good for seriation because of the difficulty of assigning them to a type and because their frequencies may vary erratically. Often such tools can be recognized because they are not carefully made in conformity with a specific model. Bad types may also occur sporadically or in such low frequency that they have little effect on a matrix. Other types, such as plain blades in the Deh Luran sequence, may not vary enough either in frequency or in style to be useful in a particular seriation. In our case, because of their great abundance and inconsistent frequency we found that these tools swamped more sensitive indicators to the detriment of our seriation. As a final example we can mention types of Susiana Buff which varied in relation to the amount of painting and size of vessels. The former factor causes this type to have two peaks in frequency; one of our types consisted of sherds of true Susiana Buff along with sherds of Susiana Black-on-Buff which did not happen to have paint on them. The amount of painting on the Black-on-Buff pots thus affected the counts of Buff.

Turning from types per se to typology, we should reiterate that typology
is not an obvious art; this is clear from the numerous publications that deal with the subject. When an archeologist begins work in an area he has no way of knowing just how the artifacts vary in time. As a consequence, if he is interested in using them for chronological purposes, he will separate as many types as possible hoping that some of the minor variations are related to time.

Splitting is not an unmixed blessing, however, because in a Presence-Absence matrix it will make all correlations lower. Also, if we were to use the Type-Percentage technique we would find it expedient to limit the number of types as much as possible in order to save computer time. One way out of this dilemma is to split first, then attempt to lump the items with identical distributions and/or eliminate the types that have few occurrences.

It is easy to adhere to the principle that types should be mutually exclusive if you are dealing with whole artifacts from an area that has been thoroughly studied, but constructing mutually exclusive types is not so easy when working with broken artifacts from a relatively unknown area. In the latter case, different parts of the same artifact may be classified as different types (as we did with two types of Susiana Buff). Exclusiveness is also violated if two types represent different stages in manufacture or separate parts of the same tool. For example, blade cores and blades should show the same distribution in time because they are part of the same manufacturing process. To count both in a matrix is to overemphasize the chronological implications of blades at the expense of other tools. To avoid these problems, the criteria for the types may have to be revised as studies progress.

In summary, our theoretical model of culture and culture change—which is based on the isolated, homogeneous community existing in a stable natural environment—is far too simple to be used uncritically. The archeological cultural contexts and the analytical filter through which the data are passed all introduce factors that are not specifically taken into account by this model.

**Archeological Considerations**

Independent of the theoretical considerations outlined above there are problems deriving from archeological data which have an effect on the outcome of a seriation. First, an archeological site contains only a fraction of the equipment used by prehistoric people. Decay, erosion, and mechanical destruction all tend to destroy artifacts, and an archeologist has to depend on what is left. In many cases the only available residue is not suitable for seriation. Obviously in these cases, other methods of dating must be used. A problem of much greater concern, because it may not be recognized, is the mixing of deposits as a result of faulty excavation or, more charitably, of unclear stratigraphy. At best there are only rare exceptions to the general
rule that there is always mixing of artifacts in sites that were occupied for more than one discernible period. This means that it is seldom possible to recover only contemporary artifacts from any stratum. Depending on the size of the sample of artifacts, the degree of mixing, and the seriation techniques used, this may or may not be a problem. Errors in recording provenience frequently occur when artifacts are washed, labelled, and sorted. The more times artifacts are handled the more chances there are for error. Another variable is the quality of the sample of material recovered.

Putting the theoretical concerns into actual archeological contexts we can suggest things that archeologists should look for before they attempt seriation. Perhaps most important, they should be aware of distances between sites because these may correlate highly with differences in the artifacts between the sites. Types popular in one area may not be found in another, or they may occur in quite different proportions. Where sites are not literally almost in sight of one another, it should be incumbent on an archeologist to consider seriously whether geographic variation may not be mistaken for temporal variation. One would guess that geographic factors may have been important in Belous’ data, for example. Geographic differences may occur in a very short space if sites are near the edge of a natural area, the other side of which is occupied by people engaged in different activities.

Although we do not present the data in this paper, we have worked extensively on the comparison of sites in different regions. The problem of separating geographic from temporal differences has not been difficult because the sites from each area clustered. The pattern of occurrence (where the phases clustered) was like the one we found in Tepe Sabz. The seriation was not all to the good, however, since we know from radiocarbon dates that some of the components in the various sites should have interdigitated. They did not.

Another problem that ought to be considered is that of “culture lag.” This implies that certain items in particular areas occur later or last longer than they do in others. In these cases seriation will deduce similarities that do not necessarily mean contemporaneity. For practical purposes, after an initial archeological inspection of a region, we would expect that examples of lag would become obvious and could be taken into account easily. A final problem involving differences within sites or among nearby sites should again be recognized. If there is a complex society or occupational specialization there will probably be artifacts related to certain status positions or to activities. In these cases, contemporary material might appear to be of different ages. This brings up the question of sampling again, but this particular problem is one that would likely be serious only in the beginning of work in an area.

This discussion should have indicated that even if the archeological cul-
tural situation satisfied all the ideal conditions we would wish, the archeological situation—which includes the site itself, the preservation of artifacts, and the excavation and processing—can introduce sufficient errors to affect an attempt at chronological ordering. Because of these factors we emphasize that to make such a statement as “Period A is characterized by a certain array of types which occur in a certain percentage” is only to state a fact. This fact does not necessarily imply a rule from which we can generalize a program of seriation which would be applicable in all cases.

**Mechanical Considerations**

A major part of this project has been directed toward developing a system of computer programs which will deduce chronological order from archeological data. We discuss here a set of design criteria for programs of this type. The first part of the discussion is more general, however, in that it is pertinent in any case where it is desirable to have a set of procedures, especially computer programs, for the analysis of numerical data. The second group of comments considers the question of ordering matrices for the purpose of seriation; this discussion is also pertinent to other areas and other disciplines in which the ordering of a matrix according to a set of well-defined standards arises. In the third section we discuss the development of the features contained in Permutation Search.

In this discussion we set aside questions of seriation theory and of the suitability of data. The resolution of these problems is not the function of a system designer, who should rather be concerned with carrying out operations suggested by the theories on data acquired in accord with them. Thus, for the purpose of designing a system of algorithmic procedures we should be able to assume the validity of our theoretical bases (but the design should be sufficiently flexible to allow for changes in the theory) and of our data (but not so uncritically as to preclude the inclusion of data-checking features in the system). If an algorithm is provided along with the theory, it is reasonable for the system designer to examine it critically; it is also reasonable for the system designer to set up new algorithms as they seem appropriate.

**General System Design Standards**

Here we discuss the design criteria for systems of numerical data processing. While these standards pertain particularly to systems of computer programs, they are also pertinent to nonprogrammed applications and, to some extent, to situations where the data to be processed are not strictly numeric. The standards fall into four categories: simplicity, appropriateness, convenience, and extendibility.

A system of programs should be simple, especially when the users are nonprogrammers. It is reasonable to expect all users to have the mathematical background to set up their raw data in an acceptable format and
enough detailed knowledge to appreciate the techniques involved, but it does not seem reasonable to expect each user to be a computer programmer. This standard of simplicity holds true even when the users are capable programmers; although some of them may be fluent in all techniques involved, users should not be required to understand and allow for the detailed workings of the programs.

Simplicity is also important from the programmer's standpoint. It is easier to develop a sophisticated program in the first place if it is well designed and well structured (and a well-structured, well-designed program can hardly help being easy to use) and it is easier to maintain if it is simply structured. Indeed, if the person who wrote the program is not the one who maintains it, simplicity of design may be the only way to keep the system viable. Finally, a simply-structured program system will be easy to extend, as we discuss below.

When we demand that a data-analysis system be appropriate, we mean that it should contain the features which its users are most likely to want. Ideally, it should be possible to obtain all pertinent computations in one way or another. The most commonly requested ones should be available for the least effort, but the system may be more complicated when it carries out less commonly used tasks. (For example, several requests may have to be made to the system rather than just one.) The prime requirement, of course, is that the system include all the programs which are necessary in the application for which it is intended. If the techniques provided by a system are related to other applications, it is also desirable to include any simple facilities which may help users from these other applications.

For the convenience of the user (who should not be expected to go to great lengths to put his data into acceptable format), a system should provide a variety of simple input formats. This is especially important if the same data are likely to be submitted to a number of different programs. The output forms should include all the information normally required in the intended application, and should also allow a variety of forms for each. If pictorial displays are pertinent (some usually are) and can be generated by a computer, they are well worth including.

By convenience, we also mean flexibility, and to some extent control. The user should be the master of the program and have control over all variable parts of his computation. An important aspect of this control is a simple (!) means for selecting which options or computations are to be performed, and in what order. Ideally, the options should provide complete coverage of the points at which the program makes arbitrary choices on procedural questions (for example, whether to single space or double space the output). For consistency with the desideratum of simplicity, this control should of course take the form of an option to change an assumed choice rather than a requirement that all choices be specified for every run.
Moreover, the notation should be such that the natural way to make a statement or request is the correct form. Another aspect of designing and programming for convenience is including programs to do things the user could do for himself but the program can do more easily—such as allowing the user to reprocess the original input (presumably in a different manner) without inputting it again.

Finally, an aspect of convenience that is becoming increasingly popular is an arrangement for dynamic, conversational control between the user and the program. There are many applications in which this is useful, and a number in which it is almost essential for efficient use of both the man’s and the machine’s time. We found in our work with PHOENIX and the Permutation Search technique that we got results faster and learned more about our data when we worked with direct control over the computation.

The final criterion we set up is extendibility. There are three aspects of extendibility that require comment. First, and most important, is the possibility that after the system has been used for a time there will be a demand for new features. It is not relevant whether these demands arise from the development of new computing techniques, from developments or corrections in the theory, from interest in the system on the part of a different set of users from the one originally intended, or from the discovery of shortcomings in the original system. It is fairly common experience that systems encounter a demand for new features at some point in their useful lives. A second aspect of extendibility is the possibility that the program will be moved to a different computer, or that new input or output devices useful to the system will be added to the computer on which it runs. In such cases, it will be easier to move the system or to add new output procedures if it was originally designed for extendibility. Finally, sophisticated users may wish to add new programs or features on their own, either to investigate new techniques or to tailor the system to their own requirements. A modicum of extra effort in the design of a system may save such users a great deal of time and effort at a later date.

A prime example related to the second and third aspects of extendibility is the use of the PHOENIX system as a tool for evaluating techniques of seriation in order to develop the theory. As noted above, this study would not have been possible without access to a high-speed digital computer; it is also true that the study would have been significantly more difficult without a system designed with extendibility as a prime criterion.

PHOENIX: A System for Automatic Seriation

For purposes of ordering matrices, we have developed a system which, we believe, amply meets the standards of design set forth above. The PHOENIX system is an abstract design goal, and two actual realizations (PHOENIX-I at Rice University and PHOENIX-II at Carnegie-Mellon Univer-
sity) presently exist. A description of PHOENIX-II intended for users is
given in Appendix C; the actual program listing is in Appendix D; and
Appendix E contains some comments on the organization of the program
and changes necessary to convert it to another computer.

The process of using PHOENIX to operate on a set of data is quite
simple. PHOENIX-II requires only a list of the operations to be per-
formed and the data, if any, needed to carry them out. Although there has
not been enough demand for PHOENIX for the system to be released to
users who are not themselves programmers, we feel that its organization
will make the system easy for them to use.

There are three classes of control instructions. These cover (1) input,
including form of data, where to put it, what to compute from it, and how to
rearrange it; (2) computation, including selection of one or more seriation
techniques and options within each; and (3) output, including selection of
tables to print, selection of graphs to print, and control over printing for-
mat. The control instructions are all 4-character commands with a high
degree of mnemonic value: ‘IRAW’ is the command for taking Input in the
form of RAW data counts and ‘PSCH’ is the command which asks for a
seriation by Permutation SearCH.

PHOENIX-I runs on a computer where the normal mode of operation
is for the programmer to stand at the console to control the program; as
a result, the operating procedure for this version of PHOENIX involves
calling for program options from the console. We feel that its control struc-
ture is not as well designed as that for PHOENIX-II, but in spite of this
relative deficiency we have had good success in having data cases run by
an operator with only written instructions and no prior knowledge of the
system.

In order to be appropriate to the task of ordering matrices, a system of
programs must provide a certain minimum of facilities. These include set-
ting up matrices for ordering, performing the orderings, and printing or
plotting the results.

PHOENIX-II provides ten different options for preparing a correlation
matrix for seriation. They request the system to (1) read a set of raw
data, construct a table of percentages, and compute a correlation matrix;
(2) read a set of percentages and compute a correlation matrix; (3) read
a precomputed correlation matrix; (4) read only a new starting order for
the existing data; (5) do precisely nothing—that is, move on to the order-
ing stage with the last set of data in the last order found; (6) revert to the
original order on the last data set; (7) lump sites in the raw data by adding
the counts of types in the specified sites and recomputing percentage and
correlation matrices; (8) combine data sets: e.g., add new sites to a pre-
vious data set by inputting more data (either percentages or raw data as
specified); (9) compute the correlation matrix with the Brainerd-Robinson
correlation coefficients (if no form for the correlation matrix is specified, this one is assumed); and (10) compute the correlation matrix with presence-absence correlation coefficients. As an additional option, we plan to add a request for a random ordering of the existing matrix as suggested by Kuzara, Mead and Dixon (1966).

We feel that this is an adequate set of tasks for our purposes; it would not, however, be difficult to add more. There are several candidates for additional tasks. One is a choice of formats for the data cards; this exists in PHOENIX-I and will be added to PHOENIX-II if we should ever need it. Two other possibilities are an option for combining types in the same way that we can now combine sites, and a similar option for adding new types. That these have not yet been installed is a consequence of our data format: changing the configuration of sites involves retyping the entire data set, but combining or adding types involves making changes in only a few cards. Tasks could also be added for normalizing data before or after computing percentages; several were available for this purpose in PHOENIX-I, but they were rarely used and we have not yet added them to PHOENIX-II. Automatic computation of the Error Coefficient is yet another possible feature.

An assortment of seriation techniques, mostly related to ordering matrices, are provided in the system. In this regard, PHOENIX includes computation facilities which are useful even though they are not directly related to the main purpose of the system. Simple option calls can select the techniques of Meighan, Ascher, Dempsey-Baumhoff, and Permutation Search, or can select no ordering. This last possibility allows the system to be used for its explicit ordering or output procedures alone—for example, we have used PHOENIX to generate contour plots of data not related to seriation. The details of selecting ordering options are given in Appendix C.

The final task we require of the system is control over the printing of results. PHOENIX-II contains three types of requests for tabular listing of results and two types of control over printing format, and allows for three types of tabular result listings (raw data, percentages, and correlation matrix), two types of printing format control (double spacing and short lines for use with the conversational consoles), and two requests for graphic display of the results (contour plot of correlation matrix and type-percentage plot). A printer is the only convenient output device available to PHOENIX-II, but PHOENIX-I has access to an oscilloscope as well as a printer. As a direct result of the simplicity of the system design, we were able to substitute a new plotting program for one of the programs in PHOENIX-I and obtain contour plots on the oscilloscope (Figs. 3-16, 3-18). It was only slightly more complicated to generate the stereographic plots (Figs. 3-17, 3-19).
We feel that the program options and organization as described above are complete, appropriate to the application, extendible, and simple to use. The flexibility and extendibility of the system are illustrated at several points, but perhaps best in the implementation of Permutation Search, as described below.

*The Permutation Search Technique: A Planned Program*

When we developed Permutation Search, we were as much interested in developing the theory of seriation as we were in actually obtaining orderings for data sets. As a consequence, we placed a heavy emphasis on building a flexible, extendible system. The various conceptual segments of the technique were implemented as separate programs and combined in such a way as to make alteration of the theory and substitution of patterns as simple as possible.

If the user calls for a seriation by Permutation Search without specifying any of the auxiliary options, he obtains as many cycles of the pattern described in Chapter II (pairwise interchange followed by successive rotation) as it takes to find no further improvements; the order and current norm are printed at the beginning and at the end of each cycle. Running with PHOENIX-II, the user may add a single option to restrict the search to either the patterns used with the immediate improvement strategy or to the delayed improvement strategy. There are also single options for selecting the patterns to use for generating permutations: these include requests for interchanges and rotations separately, and for patterns provided by the individual user. In addition to provisions for controlling the searching strategy, Permutation Search provides two options which are useful for examining and comparing the behavior of various permutation-generating patterns. The first calls for printing each improved order; the second calls for printing the action which generated the permutation and the trial norm for each permutation examined. The latter was used to generate the example of Permutation Search in Figure 2-1. One final option is provided to convert the Permutation Search technique to the Type-Percentage technique. This can be done by simply changing the norm used in the search.

Since PHOENIX-I is a conversational, interactive system, it also includes a dynamic control option which permits the user to terminate the search at any time. This will be added to PHOENIX-II when it seems appropriate.

*Open Questions and Suggestions for Further Study*

A number of interesting issues came up in the course of this study. We resolved most of those which bore directly on our work but we have not been able to follow up on all of them. The questions concern the data used in seriation, the choice of alternatives in the ordering procedures, the potential of methods of automatic seriation which are conceptually dif-
ferent from those considered in detail here, and the possible uses of graphic displays to aid in understanding rates of change.

1. As we explained in Chapter II, we have used only one matrix norm, but since no one norm is necessarily always the best it would be interesting to determine in a series of tests what effect changing the norm would have on the seriation. Kuzara, Mead and Dixon (1966) say they used two norms, but they report the results of just one. Any testing along these lines also ought to consider the effects that varying the norms have on the potential of data for seriation.

2. A related concern is the use of any particular correlation coefficient. Since the coefficient determines the aspects of the differences among sites that will be emphasized in the seriation, more testing and evaluating seems warranted.

3. We have offered substantial evidence to show the effects of different techniques for ordering matrices, but more testing should be done on variations of the search patterns we used in Permutation Search. We used two patterns but there are many more possible forms.

4. We are still evaluating the Type-Percentage technique but the work we have done has not resolved the question of which norm to use, how to find the true peak for each type, and how to select types to use with the technique.

5. We badly need an a priori method of evaluating the potential of data for seriation. We have had good results using Meighan's 3-pole technique but it seems more fruitful in the long run to begin by considering particular sets of data and attempting to gain an understanding of the processes of change they exhibit. Here it may be necessary to base our models on studies of modern-day peoples where causal factors can be controlled.

6. Any investigations of the characteristics of data sets will also have to consider the nature of typology, specifically as it applies to seriation.

7. If through careful testing (and perhaps study of modern situations) we can gain an understanding of the characteristics of acceptable data sets, then we ought to be able to apply that knowledge to data sets of unknown quality. This might be done prior to or during the process of seriation so that we can tell how the data are behaving. In these cases, we should then be able to revamp the data set and re-seriate the improved version with some confidence.

8. Another approach to improving a data set would be to scale or weight different types. A blind weighting of types, to counteract great differences in frequency, for example, would probably be inadvisable. On the other hand, some weighting based on archeological information may be possible.

9. Meighan's technique deserves further exploration to include an arbitrary number of types. Since the method is independent of input order, it is an attractive prospect to extend it although its realization is not in hand. We
conjecture that the technique can be formulated as a line-fitting problem in 3-space (instead of 2-space as Ascher [1959] treats it), and that it can then be generalized into n-space. Such a generalization would lead fairly directly to a technique which would give the best straight-line fit to all the points; and a numerical indication of how good the approximation is. A graphic display which would allow an intuitive grasp of the results is not, however, so simple.

10. A final consideration is in the interpretations that may be made from contour plots. We have shown some of the characteristics of data that can be deduced from the plots but it is safe to say that much more might be learned. In particular there are two problems that contour plots might help solve. First, estimating the magnitude of a break in a sequence by the sharpness of the break in the contour seems interesting. Second, a determination of rates of change may be gained by observing the slope of the contours and the between-column errors. In this regard, refer back to Figure 3-20 in which a cross section through a contour plot shows a series of steps. Visualizing these as curves of growth or rates of change is not too difficult although it remains to be tested whether we can do it from the data we now use. To determine rates, a control of time as well as magnitude is needed. In any case, a comparison of different data sets from the same series of sites give dramatic evidence of relatively different rates of change among data sets. These data, graphically expressed, could go a long way toward making the nature of change more understandable.

The foregoing questions stress the need for further testing of techniques and hypotheses. Very little of the archeological data we now have, however, is really suitable for use in these tests. It should be stressed, therefore, that we can approach some archeological field situations as if they were testing grounds for certain techniques and hypotheses and we are not going to get suitable data unless we do. A good bit of such data gathering can go on without in any way affecting the other aspects of archeological research—except that it may help a person understand his site better if he occasionally digs it and analyzes it in new ways.
APPENDIX A

TABLES OF RAW DATA
DATA SET I. DEH LURAN POTTERY

Type
Jaffar Painted
  1 Convex-walled bowls
Jaffar Plain
  2 Rims of deep, open bowls
  3 Bases of deep, open bowls
  4 Convex-walled bowls
  5 Hole-mouth jars
  6 Dimple bases
Khazineh Red
  7 Open bowls
  8 Hemispherical bowls
  9 Hole-mouth jars
  10 Necked jars
  11 Dimple bases
  12 Carinations
  13 Flat bases
Susiana Black-on-Buff
  14 Basins
  15 “Type” 14
  16 “Type” 15
  17 “Type” 16
  18 Bases of “Type” 14 or 16
  19 Goat bowls
  20 Open-mouth “Type” 1
  21 Close-mouth “Types” 2, 3, 6
  22 Carinated “Type” 13
  23 “Type” 11
  24 “Type” 11a
  25 “Type” 12
  26 “Type” 12a
  27 Deep bowls with flaring rims
  28 High-necked jars
  29 Short-necked jars
  30 Medium-necked jars
  31 Sherds of jars
  32 Ledge-rim jars
  33 Pedestals
  34 Shallow bowls with pedestals
  35 Ring bases
  36 Sauce boats
Susiana Buff
  37 Rims of basins
  38 Bases of vertical-sided basins
  39 Bases of open bowls or basins
  40 Rims of deep, open bowls
  41 Bases of open carinated bowls
  42 Rims of deep, bell-shaped bowls
  43 Rims of deep, straight-sided bowls
  44 Sabz bowls
  45 Rims of hemispherical bowls
  46 Rims of hole-mouth jars
  47 Dimple bases
  48 High-necked jars
  49 High or low-neck jars
  50 Ledge-rim jars
  51 Pedestal bases
  52 Ring bases
Mehmeh Red-on-Red
  53 Open, straight-sided bowls
  54 Hole-mouth jars
Bayat Red
  55 Carinated bowls
  56 Hole-mouth jars
  57 Necked jars
### Computer Analysis of Chronological Seriation

<table>
<thead>
<tr>
<th>DATA SUBSET</th>
<th>DATA</th>
<th>RAW DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaffar Painted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jaffar Plain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Khuzestan Red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Susiana Black-On-Buff (ib)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Susiana Buff (ib)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Khuzestan Red-On-Red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayat Red (ic)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Trapezes</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Triangles</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Crescents</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Diagonal-ended microliths</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Backed and diagonal-ended microliths</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Backed bladelets</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Drills</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Reamers</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Flakes with ends retouched on the bulbar side</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Pointed pieces</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Plain sickles</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Used blades</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Used flakes</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Plain blades</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Blade, round-ended scrapers</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Blade, miscellaneous-ended scrapers</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Blades, with bulbar side-end retouch</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Flake, round-ended scrapers</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Steep, rough scrapers</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Flakes with bulbar-end retouched</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Thin sections</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Pressure flaked pieces</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Blunted pieces</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Micro-burins</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Burins</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Fabricators</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Scaled pieces</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Nibbled bladelets</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Truncated/backed sickles</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Backed blades</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Truncated/backed blades</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Bifacially-chipped flakes</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Bifacially-chipped points</td>
<td></td>
</tr>
</tbody>
</table>

(Abbreviated Listing)

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(4,5)  Diagonal-ended, and backed and diagonal-ended microliths</td>
</tr>
<tr>
<td>2</td>
<td>(6)  Backed bladelets</td>
</tr>
<tr>
<td>3</td>
<td>(7,8) Drills and reamers</td>
</tr>
<tr>
<td>4</td>
<td>(11)  Plain sickles</td>
</tr>
<tr>
<td>5</td>
<td>(15,16,18) Blade, round-ended scrapers; blade, miscellaneous-ended scrapers; and flake, round-ended scrapers</td>
</tr>
<tr>
<td>6</td>
<td>(28)  Nibbled bladelets</td>
</tr>
<tr>
<td>7</td>
<td>(29)  Truncated/backed sickles</td>
</tr>
<tr>
<td>8</td>
<td>(30,31) Backed blades, and truncated/backed blades</td>
</tr>
<tr>
<td>9</td>
<td>Bullet cores</td>
</tr>
<tr>
<td>10</td>
<td>Semi-chipped blade cores</td>
</tr>
<tr>
<td>11</td>
<td>Semi-chipped flake cores</td>
</tr>
<tr>
<td>12</td>
<td>Amorphous cores</td>
</tr>
<tr>
<td>13</td>
<td>Thin nodule cores</td>
</tr>
</tbody>
</table>
### Abbreviated Listing

<table>
<thead>
<tr>
<th>Type</th>
<th>Raw Data</th>
<th>BUS 2</th>
<th>BUS 1 AL12</th>
<th>AL11</th>
<th>MD2</th>
<th>MD1</th>
<th>512 Z</th>
<th>KHA3</th>
<th>KHA2</th>
<th>KHA1</th>
<th>MHE2</th>
<th>MHE1</th>
<th>MHE</th>
<th>MUSE</th>
<th>RA62</th>
<th>RAY2</th>
<th>RAY1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>COM</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>ypc</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>51</td>
<td>AM</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3 PT A</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>PUS</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>ALI1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>YHA</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL12</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL11</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table

<table>
<thead>
<tr>
<th>Type</th>
<th>Raw Data</th>
<th>BUS 2</th>
<th>BUS 1 AL12</th>
<th>AL11</th>
<th>MD2</th>
<th>MD1</th>
<th>512 Z</th>
<th>KHA3</th>
<th>KHA2</th>
<th>KHA1</th>
<th>MHE2</th>
<th>MHE1</th>
<th>MHE</th>
<th>MUSE</th>
<th>RA62</th>
<th>RAY2</th>
<th>RAY1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>COM</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>ypc</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>51</td>
<td>AM</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3 PT A</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>PUS</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>ALI1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>YHA</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL12</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>AL11</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### DATA SET III. PA SANGAR FLINT

( Abbreviated list from areas “a,” “b,” “c”)

<table>
<thead>
<tr>
<th>Type</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(10) Arjeneh points</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>(5,6,7,11,12) Various retouched blades</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>(2) Diagonal-ended bladelets</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>(8) Backed bladelets</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>(16,19) Notched blades and flakes, used blades</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
</tr>
</tbody>
</table>

(Complete list from area “a”)

<table>
<thead>
<tr>
<th>Type</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Crescents</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>Diagonal-ended bladelets</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>Backed and diagonal-ended bladelets</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>Triangles</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>Baradostian bladelets “Type 1”</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>Baradostian bladelets “Type 2”</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>Baradostian bladelets “Type 3”</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>Backed bladelets</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>Backed bladelet fragments</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>Arjeneh points</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>Blades and bladelets (TT)</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Baradostian blades (B)</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>Mousterian points</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>Reamers</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>Pointed pieces</td>
<td>33</td>
</tr>
<tr>
<td>16</td>
<td>Used blades</td>
<td>34</td>
</tr>
<tr>
<td>17</td>
<td>Used flakes</td>
<td>35</td>
</tr>
<tr>
<td>18</td>
<td>Plain blades</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>38</td>
</tr>
</tbody>
</table>
(Abbreviated list from areas “a,” “b,” “c”)
from area “a”)

<table>
<thead>
<tr>
<th>Type</th>
<th>P51</th>
<th>P512</th>
<th>P513</th>
<th>P514</th>
<th>P515</th>
<th>P516</th>
<th>P517</th>
<th>P518</th>
<th>P519</th>
<th>P520</th>
<th>P521</th>
<th>P522</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
DATA SET IV. 110 TYPES

Type
Spindle Whorls
1 Perforated stone discs
2 Perforated sherd discs
3 Star-shaped whorls
4 Chariot-wheel whorls
5 Oval-discoidal whorls
6 Punctuate-dome-shaped whorls

Chipped Sherds
7 Hoe-shaped
8 Discoidal
9 Miscellaneous

Ceramic Artifacts
10 Bent nails
11 Chipped sherds
12 Pot lids
13 Ceramic cone
14 Toy tops
15 Sling missiles

Grinding Slabs
16 Flat-topped boulder
17 Saddle-shaped
18 Shallow basin
19 Combination shallow basin and mortar
20 Pebble mortar
21 Boulder mortar
22 Combination saddle-shaped and mortar
23 Saddle-shaped with central depression
24 Miscellaneous fragments

Handstones, Pestles, and Pounders
25 Simple discoidal
26 Loaf-shaped
27 Irregular elongate
28 Small slab abrader
29 Irregular sausage-shaped
30 Conical pestle and rolling handstone
31 Cylindrical pestle and rolling handstone
32 Stubby bell-shaped muller
33 Irregular sausage-shaped pestle
34 Core pestle
35 Spherical pounder
36 Core pounder
37 Cuboid pounder

Miscellaneous Stone Tools
38 Pebble choppers
39 Chipped stone hoes
40 Polished celts
41 Picks
42 Grooved mauls
43 Chisels
44 Sashweights
45 Slicing slabs
46 Grooved rubbing stones
47 Small, faceted rubbing stones
48 Chipped limestone discs
49 Perforated stones
50 Cleaver-rubbing stone
51 Limestone discs
52 Large basin with lid
53 Pecked stone balls
54 Grooved stone balls
55 Incised pebbles
56 Grooved amulets
57 Stone basin with handle
58 Fragments of polished stone
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION

Figurines
59 Animal
60 Human
61 Horn-shaped
62 T-shaped
63 Clay hand
64 Other fragments

Miscellaneous Objects of Clay
65 Clay cylinders with pinched ends
66 Clay cylinders with flared ends
67 Clay cylinders with applications or punctations
68 Clay cylinder fragments
69 Clay stalks
70 Clay balls
71 Clay bases
72 Ovoid clay lumps
73 Clay polisher
74 Grooved polisher

Mats, Baskets and Textiles
75 Over-one, under-one mats
76 Over-two, under-two mats
77 Over-one, under-one baskets
78 Coiled baskets
79 Plain weave textiles

Bone Artifacts
80 Awls
81 Needles
82 Spatulas
83 Chisel
84 Gouge
85 Pressure-flaker
86 Handles
87 Bone haft
88 Knife
89 Polished bone
90 Cut long bones

Ornaments
91 Boar’s tusk pendant
92 Bell-shaped pendant
93 Flat pebble pendant
94 Clam shell pendant
95 Plummets-shaped pendant
96 Miscellaneous stone pendants
97 Miscellaneous bone pendants
98 Crab claw pendant
99 Cuff link labret
100 T-shaped labret
101 Nail-shaped labret
102 Turquoise labret
103 Smooth-surface bracelet
104 Incised bracelet
105 Clam-shell button
106 Boar’s tusk button
107 Ceramic rings
108 Ring blanks
109 Stone spool
110 Copper pin
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>
APPENDIX B

COMPLETE OUTPUT FOR DATA SUBSET Ia
Example of Computer Output for Seriation of Data Subset Ia (Susiana Black-on-Buff Pottery)
### COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION 113

#### DATA SUBSET IA (SUSIANA BLACK-ON-BUFF POTTERY)

<table>
<thead>
<tr>
<th></th>
<th>HA3</th>
<th>HA2</th>
<th>HA1</th>
<th>MEW3</th>
<th>MEM2</th>
<th>MEM1</th>
<th>MUSE</th>
<th>BAY3</th>
<th>BAY2</th>
<th>BAY1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SABZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON LOOP 0 VALUE IS 333.3 FOR ORDER AND COLUMN ERRORS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SABZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON LOOP 1 VALUE IS 193.2 FOR ORDER AND COLUMN ERRORS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SABZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON LOOP 2 VALUE IS 193.2 FOR ORDER AND COLUMN ERRORS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SABZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CORRELATION MATRIX

<table>
<thead>
<tr>
<th></th>
<th>HA3</th>
<th>HA2</th>
<th>HA1</th>
<th>MEW3</th>
<th>MEM2</th>
<th>MEM1</th>
<th>MUSE</th>
<th>BAY3</th>
<th>BAY2</th>
<th>BAY1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SABZ</td>
<td>203.0</td>
<td>110.1</td>
<td>58.1</td>
<td>54.6</td>
<td>57.3</td>
<td>53.1</td>
<td>71.1</td>
<td>87.6</td>
<td>80.5</td>
<td>79.4</td>
</tr>
<tr>
<td>51.1</td>
<td>200.0</td>
<td>102.2</td>
<td>104.4</td>
<td>109.8</td>
<td>77.5</td>
<td>94.4</td>
<td>89.1</td>
<td>84.6</td>
<td>79.4</td>
<td>73.5</td>
</tr>
<tr>
<td>54.6</td>
<td>102.2</td>
<td>200.0</td>
<td>184.9</td>
<td>124.9</td>
<td>81.0</td>
<td>77.4</td>
<td>73.0</td>
<td>57.9</td>
<td>57.8</td>
<td>53.5</td>
</tr>
<tr>
<td>57.3</td>
<td>107.8</td>
<td>124.9</td>
<td>140.4</td>
<td>205.0</td>
<td>137.9</td>
<td>144.7</td>
<td>138.6</td>
<td>129.8</td>
<td>129.1</td>
<td>110.4</td>
</tr>
<tr>
<td>53.4</td>
<td>77.5</td>
<td>81.6</td>
<td>94.2</td>
<td>137.3</td>
<td>200.0</td>
<td>150.3</td>
<td>158.5</td>
<td>116.9</td>
<td>114.8</td>
<td>115.0</td>
</tr>
<tr>
<td>71.1</td>
<td>94.4</td>
<td>77.4</td>
<td>94.1</td>
<td>144.7</td>
<td>166.3</td>
<td>200.0</td>
<td>168.3</td>
<td>131.1</td>
<td>139.8</td>
<td>135.7</td>
</tr>
<tr>
<td>87.6</td>
<td>99.1</td>
<td>73.0</td>
<td>90.5</td>
<td>124.8</td>
<td>194.5</td>
<td>158.3</td>
<td>200.0</td>
<td>190.6</td>
<td>153.6</td>
<td>143.8</td>
</tr>
<tr>
<td>98.9</td>
<td>84.0</td>
<td>57.9</td>
<td>73.7</td>
<td>129.1</td>
<td>114.9</td>
<td>131.1</td>
<td>153.6</td>
<td>290.0</td>
<td>164.8</td>
<td>167.0</td>
</tr>
<tr>
<td>76.4</td>
<td>79.4</td>
<td>57.6</td>
<td>73.9</td>
<td>100.0</td>
<td>114.8</td>
<td>139.0</td>
<td>150.9</td>
<td>164.6</td>
<td>208.8</td>
<td>178.8</td>
</tr>
<tr>
<td>73.7</td>
<td>73.5</td>
<td>53.3</td>
<td>69.4</td>
<td>94.0</td>
<td>113.8</td>
<td>135.7</td>
<td>143.8</td>
<td>167.0</td>
<td>178.8</td>
<td>200.0</td>
</tr>
</tbody>
</table>

### NORM 13 193.2 FOR ORDER, COLUMN ERRORS, AND INTER-COLUMN ERRORS

<table>
<thead>
<tr>
<th></th>
<th>HA3</th>
<th>HA2</th>
<th>HA1</th>
<th>MEW3</th>
<th>MEM2</th>
<th>MEM1</th>
<th>MUSE</th>
<th>BAY3</th>
<th>BAY2</th>
<th>BAY1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAW DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SABZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NORM 13 193.2 FOR ORDER, COLUMN ERRORS, AND INTER-COLUMN ERRORS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SABZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAW DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SABZ</td>
<td>39.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9.0</td>
<td>10.0</td>
<td>53.0</td>
<td>105.0</td>
<td>55.0</td>
<td>4.0</td>
<td>12.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3.1</td>
<td>2.0</td>
<td>9.5</td>
<td>10.0</td>
<td>47.0</td>
<td>6.6</td>
<td>31.0</td>
<td>25.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.1</td>
<td>4.0</td>
<td>8.0</td>
<td>9.5</td>
<td>39.0</td>
<td>2.0</td>
<td>16.0</td>
<td>10.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>13.0</td>
<td>10.0</td>
<td>33.0</td>
<td>35.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>15.0</td>
<td>2.0</td>
<td>23.0</td>
<td>27.0</td>
<td>14.0</td>
<td>1.0</td>
<td>7.0</td>
<td>12.0</td>
<td>3.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>43.0</td>
<td>14.0</td>
<td>5.0</td>
<td>4.0</td>
<td>9.0</td>
<td>1.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3.0</td>
<td>3.0</td>
<td>29.0</td>
<td>18.0</td>
<td>20.0</td>
<td>3.0</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>4.0</td>
<td>5.0</td>
<td>4.0</td>
<td>21.0</td>
<td>20.0</td>
<td>156.0</td>
<td>130.0</td>
<td>129.0</td>
<td>101.0</td>
<td>55.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>19.0</td>
<td>12.0</td>
<td>37.0</td>
<td>31.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>19.0</td>
<td>12.0</td>
<td>37.0</td>
<td>31.0</td>
</tr>
<tr>
<td>233.0</td>
<td>42.0</td>
<td>45.0</td>
<td>78.0</td>
<td>60.1</td>
<td>13.0</td>
<td>131.0</td>
<td>147.0</td>
<td>150.0</td>
<td>86.0</td>
<td>48.0</td>
</tr>
<tr>
<td>162.0</td>
<td>5.0</td>
<td>1.0</td>
<td>9.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>2.0</td>
<td>7.0</td>
<td>13.0</td>
<td>13.0</td>
<td></td>
</tr>
<tr>
<td>15.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>2.0</td>
<td>9.0</td>
<td>26.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td>0.0</td>
<td>7.0</td>
<td>15.0</td>
<td>11.0</td>
<td>1.0</td>
<td>3.0</td>
<td>5.0</td>
<td>10.0</td>
<td>6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5.0</td>
<td>0.0</td>
<td>2.0</td>
<td>2.0</td>
<td>6.0</td>
<td>3.0</td>
<td>4.0</td>
<td>7.0</td>
<td>20.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>9.0</td>
<td>2.0</td>
<td>1.0</td>
<td>4.0</td>
<td>1.0</td>
<td>4.0</td>
<td>9.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>9.0</td>
<td>2.0</td>
<td>1.0</td>
<td>4.0</td>
<td>1.0</td>
<td>4.0</td>
<td>9.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>HA2</td>
<td>KA2</td>
<td>KA1</td>
<td>ME2</td>
<td>MUSE</td>
<td>ME1</td>
<td>BAY2</td>
<td>BAY3</td>
<td>BAY1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>8.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.3</td>
<td>10.5</td>
<td>22.7</td>
<td>27.0</td>
<td>14.7</td>
<td>5.9</td>
<td>5.2</td>
<td>2.8</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.4</td>
<td>2.1</td>
<td>0.4</td>
<td>2.6</td>
<td>12.6</td>
<td>0.8</td>
<td>11.4</td>
<td>4.7</td>
<td>1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.7</td>
<td>3.5</td>
<td>20.2</td>
<td>23.6</td>
<td>4.6</td>
<td>2.9</td>
<td>3.6</td>
<td>2.4</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.5</td>
<td>14.7</td>
<td>7.4</td>
<td>8.9</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>2.1</td>
<td>10.7</td>
<td>5.9</td>
<td>4.3</td>
<td>1.5</td>
<td>1.6</td>
<td>2.8</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.2</td>
<td>14.9</td>
<td>2.1</td>
<td>1.0</td>
<td>6.9</td>
<td>1.5</td>
<td>0.4</td>
<td>0.7</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.4</td>
<td>3.2</td>
<td>12.4</td>
<td>4.6</td>
<td>6.4</td>
<td>3.0</td>
<td>6.4</td>
<td>3.0</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>4.3</td>
<td>3.4</td>
<td>10.3</td>
<td>21.7</td>
<td>29.4</td>
<td>35.0</td>
<td>31.1</td>
<td>28.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.0</td>
<td>2.0</td>
<td>6.0</td>
<td>4.3</td>
<td>26.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.4</td>
<td>44.7</td>
<td>19.3</td>
<td>20.1</td>
<td>21.4</td>
<td>19.1</td>
<td>26.4</td>
<td>34.7</td>
<td>33.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29.2</td>
<td>3.2</td>
<td>0.4</td>
<td>0.0</td>
<td>8.0</td>
<td>3.3</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.5</td>
<td>0.0</td>
<td>6.2</td>
<td>0.3</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.8</td>
<td>0.3</td>
<td>6.9</td>
<td>3.2</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.3</td>
<td>0.0</td>
<td>6.9</td>
<td>3.2</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.8</td>
<td>0.0</td>
<td>3.0</td>
<td>7.9</td>
<td>3.9</td>
<td>15.9</td>
<td>6.7</td>
<td>1.2</td>
<td>1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.6</td>
<td>4.4</td>
<td>6.9</td>
<td>1.7</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.7</td>
<td>5.3</td>
<td>0.9</td>
<td>0.3</td>
<td>1.1</td>
<td>1.5</td>
<td>6.2</td>
<td>0.3</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

114 RICE UNIVERSITY STUDIES
APPENDIX C

USER’S GUIDE TO PHOENIX-II
Introduction

PHOENIX-II is a general system incorporating a variety of matrix ordering techniques and input-output controls. Input may be raw data, a percentage table, or a precomputed correlation matrix. Input features include specification of an order for the data set currently in the machine, addition or deletion of sites, lumping of sites, and (potentially) specification of ordering constraints. If the input is raw data or a percentage table, a correlation matrix is computed automatically. The user may select whether this matrix is computed according to the Brainerd-Robinson algorithm or as a "presence-and-absence" matrix after Dempsey and Baumhoff.

After inputting his data, the user has a choice of the ordering technique to be used on it. At present, the Permutation Search and Type-Percentage techniques are the only ones available; there are provisions for adding more, and we plan to add Meighan's, Dempsey and Baumhoff's, and Ascher and Ascher's. Each of the ordering subprograms has a number of internal options controlling such things as printer output and search pattern.

After each application of an ordering program, the user has his choice of (1) applying another program to the newly-found order, (2) returning to the input order and applying another program, (3) specifying a new input order and applying another program, (4) reading a completely new data set, (5) printing some information about the order just found and returning to select one of the other alternatives, or (6) stopping.

The output available at present includes decimal printouts of the correlation matrix, the raw data, and the percentage table (which is computed and saved if raw data is input), contour plots of the correlation matrix, and type-percentage graphs (battleship plots) of the percentage table.

Methods

Several approaches to seriation are included in the PHOENIX system. Most are based on the analysis of a correlation matrix derived from tabulations of artifact types found in the sites; the others work directly on the percentage tabulations.

Correlation Matrices: Two forms are available and more can be added without difficulty.

Let 

- \( P \) = number of sites
- \( T \) = number of types
- \( A_{i,j} \) = percentage of site \( J \) composed of type \( I \)
- \( X_{i,j} \) = correlation of site \( I \) with site \( J \)
then:

**Brainerd-Robinson Matrix.**
[Proposed by Robinson (1951)]

\[
X_{I,J} = 200.0 - \sum_{K=1}^{T} |A_{K,I} - A_{K,J}|
\]

Note: \(X_{I,I} = 200.0\)

**Presence-and-Absence Matrix.**
[Proposed by Dempsey and Baumhoff (1963)]

\[
X_{I,J} = \sum_{K=1}^{T} \begin{cases} 
0 & \text{if } A_{K,I} \times A_{K,J} = 0 \text{ and } A_{K,I} + A_{K,J} \neq 0 \\
1 & \text{if } A_{K,I} \times A_{K,J} \neq 0 \text{ or } A_{K,I} + A_{K,J} = 0
\end{cases}
\]

That is, this computes a tally of "common responses" (both types present or both types absent) between sites \(I\) and \(J\).

Note: \(X_{I,I} = T\)

**Ordering Methods:** Four ordering techniques are included in the design and more can be added without difficulty.

*Permutation Search.* This is the mainstay of the PHOENIX system. The method, described elsewhere, is a new attempt to deduce chronology by the criteria of Brainerd and Robinson.

*Type-Percentage.* This is an outgrowth of the Permutation Search technique. The method attempts to derive an ordering from characteristics of the percentage matrix rather than from the correlation matrix.

*Dempsey-Baumhoff.* Contextual analysis as proposed by Dempsey and Baumhoff (1963) is included as far as it can be programmed algorithmically.

*Ascher.* This section is a transliteration and slight generalization (to handle varying input orders) of Ascher and Ascher's (1963) attempt to order by the criteria of Brainerd and Robinson.

The *Meigham* method of 3-pole plots (1959) has not yet been programmed. When it is added as an option, it should include an extension to handle as many types as desired, rather than the three types to which it is presently restricted.

**Norms:** One norm or "agreement coefficient" is presently available for correlation matrices. As above, more can be added or this one changed without great difficulty. In addition, a norm computed directly from the type-percentage table is available for the type-percentage facet of Permutation Search.
As above, let

\[ P = \text{number of sites} \]
\[ T = \text{number of types} \]
\[ A_{i,j} = \text{percentage of site } J \text{ composed of type } I \]
\[ X_{i,j} = \text{correlation of site } I \text{ with site } J \]

then:

**Sum-of-errors norm for correlation matrix**

\[
\text{Norm} = \sum_{i=1}^{P} \sum_{j=1}^{P-1} \left| X_{i,j} - X_{i,j+1} \right| \cdot \Delta
\]

where \( \Delta = 0 \) if the entries in the correlation matrix go the right way (i.e. ascending toward the diagonal) and \( \Delta = 1 \) if the entries go the wrong way.

**Sum-of-errors norm for type-percentage table**

\[
\text{Norm} = \sum_{i=1}^{T} \sum_{j=1}^{P-1} \left| A_{i,j} - A_{i,j+1} \right| \cdot \Delta
\]

where \( \Delta = 0 \) if the percentages go the right way (i.e. ascending toward the peak) and \( \Delta = 1 \) if the percentages go the wrong way.

**Usage**

A series of computations and attempts at seriation to be performed by PHOENIX must be presented as a sequence of tasks. Each task includes requests for input, computation, and output. For each case, the various options for the task are selected first and any data needed are supplied on following cards.

A complete run submitted to PHOENIX is a sequence of tasks which may, for example, read a data set, attempt to order it with each of three techniques used individually (one task for each), try several particular orders (one task each), try several random input orders (one task each) followed by Permutation Search, and begin again on different data.

**Information needed for Each Data Case:** Eight pieces of information are needed for each data case:

- **TITLE** An 80-character label used for printing output. It is not actually used in computation.
- **T** The number of types. Not needed if input is a correlation matrix.
- **P** The number of sites or components.
- **SITE.NAM** A list of 4-character labels—one for each site. These are not used in actual computation, but are used to label printing output.
- **RAW** A table of raw counts of occurrences of the T types in the P sites. Not needed if input is a correlation matrix or percentage table.
A A table of percentages of the T types in the P sites. It is computed automatically if input is a table of raw data; it is not needed if input is a correlation matrix.

X The correlation matrix, either read as input or computed by one of the techniques described above.

L The order in which the sites have been placed. Initially, this is the same order as the input data.

Options: The three classes of options available provide for reading or changing the data (input), selection and control of ordering techniques (computation), and printing or plotting of results (output). Each option is requested by punching its four-character name into an option request card as described below.

(1). Input Options. The two types of information needed to set up the data for the ordering programs are the format of the data to be read and the type of correlation matrix to be set up.

Input Data Format

IRAW Input is raw data counts of types in sites.
IPCT Input is a tabulation of percentages of types in sites.
ICOR Input is precomputed correlation matrix.
IADD The input to be read by this task (format specified by IRAW or IPCT) is to be added to the data already in the machine (as extra columns at the right of the array) instead of replacing them.
IORD No new data are to be supplied, but a new starting order for the data must be read.
ILST No new data; continue computing with the last order found on the last trial.
IORG No new data, revert to the input order for a starting order.
IGRP No new data; new cards which specify columns to be lumped together must be read.

Some discretion and common sense should be exercised here. For example, it is not meaningful to call on more than one of these input selection options for each task, with the exception of the combinations IRAW,IADD and IPCT,IADD.

Selection of Correlation Matrix

MXBR Compute correlation matrix with Brainerd-Robinson scheme. If no correlation matrix is specified, this one is assumed.
MXPM Compute correlation matrix with presence-and-absence scheme.

(2). Computation Options. The computation options are discussed in
the order in which they are processed. That is, if both ASCH and PSCH are called for, the Ascher method is carried out first, and then the Permutation Search.

Randomize Input Order
  RNDM  Generate a random starting order.

Ascher Method
  ASCH  Call on Ascher seriation program.
  ASTR  Print intermediate results for Ascher order.

Meighan Method
  MEIG  Call on Meighan program.
  MGTR  Print intermediate results for Meighan program.

Dempsey-Baumhoff Method
  DPBM  Call on Dempsey-Baumhoff program.
  DBTR  Print intermediate results for Dempsey-Baumhoff program.

Permutation Search Method
  PSCH  Call on Permutation Search program.
  PSTR  Print each order accepted as an improvement.
  PSAL  Print action and norm at each trial. (BEWARE: this one generates a lot of paper.)
  PIMM  Use the immediate-improvement pass only.
  PDLY  Use the delayed-improvement pass only.
  PSQ1  Use search pattern 1 (interchanges).
  PSQ2  Use search pattern 2 (rotations).
  PSQ3  Use search pattern 3 (not defined).
  PSQ4  Use search pattern 4 (not defined).
(Note: PSQ1, PSQ2 is normally assumed.)

Type-Percentage Method
  TPCT  Convert Permutation Search to the Type-Percentage technique—use percentage table and a different norm. Meaningful only if input data was not a correlation matrix and interpreted only if called as a suboption with PSCH. All Permutation Search Options except PSCH apply here just as for Permutation Search.

Output Options. Two categories of output are available: numerical printouts and plots. In addition, two format controls for the numerical printouts are provided.

Numerical Listings
  ORAW  Print the raw data in the most recent order found or defined.
  OPCT  Print the percentage table in the most recent order.
  OCOR  Print the correlation matrix in the most recent order.
Plots
  OCTR  Print a contour plot of the correlation matrix in the most recent order found.
  OSHP  Print a type-percentage (battleship) plot in most recent order.

Format Controls
  OTTY  Use short output line. The printer paper is 120 columns in width, but our remote teletype stations may be as narrow as 72 columns.
  ODBL  Double-space the output. This holds for numerical printouts only—not the plots.

Termination
  STOP  Leave the PHOENIX system. This task is used only to terminate the run.

**Data Formats:** All the forms in which data cases are presented to PHOENIX are described here. Note that numerical data are almost always written (punched) in a free format, just the way one would write them on paper. Each number may be typed in as many or as few spaces as desired, with or without a decimal point as common sense deems appropriate. Spaces may be inserted at will, but each number must be followed by a comma. This means that numbers in a list are separated by commas and, note, the last number of the sequence is followed by a comma.

1. **Option Card.**
   *Purpose:* To call for the options.
   *Format:* Beginning in column 1, a string of 4-character options separated by single spaces or commas. A blank option name means that no more options appear on this card but more may appear on the next; the option “name” 1111 means that there are no more options for this task. The order in which the options appear is irrelevant. Note that prose comments following the 1111 will be disregarded by the system.
   *Example:* Examples must have all characters of equal width.
   If __ denotes a blank space, the following are equivalent option cards:
   1. IRAW_PSCH_PSTR_OCOR_ORAW__
      OSHP__OTTY__1111
   2. IRAW,PSCH,PSTR,OCOR,ORAW,OSHP,OTTY,1111
   3. OTTY,PSTR__
      IRAW,PSCH,OCOR,ORAW,OSHP,1111
   4. OTTY,ORAW,PSTR,IRAW__OSHP__
PSCH__OCOR____lll THIS IS AN EXAMPLE

(2). Teletype Option Card.
Purpose: This card is used to select which part of the output being printed on the computer's line printer is also to be printed at the remote teletype station.
Format: Same as for option card. The output options are useful here; the options which normally call on ordering procedures will, if used here, cause information printed during ordering to be sent to the teletype as well.

(3). Title Card.
Purpose: The information provided here is used to label output.
Format: One 80-column card containing any information desired.

(4). P Card.
Purpose: To specify the number of sites in the data about to be read; it is used when the number of types is not relevant.
Format: One integer followed by a comma, with no restrictions on columns. (An integer is just what you think it is: a digit or string of digits without a decimal point.)
Example: Again __ represents a blank column; the following are equivalent:
1. 8,
2. __ 8,
3. __ __ 8 __,
4. 8 __ __

(5). T P Card.
Purpose: To specify the number of types and the number of sites in the data about to be read.
Format: Two integers, each followed by a comma—no restrictions on columns.
Example: The following are equivalent:
1. 8,10,
2. __ 8, __ 10,
3. 8 __ , __ 10 __,
4. __ __ 8, __ 10,

(6). Site Name Card.
Purpose: To provide both a list of names of sites and an order for the list. When a SITE NAME CARD is used along with input data, it gives the names of the sites in the order of the data. When it is used with a request for a new input order, it gives the desired new order.
Format: Four-character names, each followed by a space or a comma. If more than 16 sites are involved, the first 16 go on one card, the 17th through the 32nd go on the next card, and so forth. It is not permissible to put fewer than 16 names on any but the last of the site name cards.

Example: The following are equivalent:
AAAA__BBBB__CCCC
AAAA , BBBB , CCCC

(7). Data Array.

Purpose: To give the actual data for a correlation matrix, a percentage table, or a set of raw data.

Format: The array is read as a series of rows, each with a given number of entries. The number of rows is given by the first number on the T P CARD for raw data or percentages; it is given by the only number on the P CARD for a correlation matrix. The number of entries (numbers) in each row is given by the second number on the T P CARD for raw data or percentages; it is given by the only number on the P CARD for a correlation matrix. Each row must begin a new card. Aside from this restriction, the rows may be punched in a free format, with as many numbers per card as desired in any convenient format. Blank cards are permitted.

Example: See the complete data cases given below.

(8). Combining Card.

Purpose: To specify the sites to be lumped together and the new name for their sum.

Format: This is the only card with a rigid format for numerical data.

Example: SUMA__03__AAAA__ABCD__BBBB__

Organization of a Run: A run under PHOENIX is composed of a series of TASKs followed by a STOP TASK. Thus:
A TASK is composed of an OPTION CARD followed by a TELETYPE OPTION CARD followed by a DATA CASE. Thus:

- OPTION CARD
- TELETYPE OPTION CARD
- DATA CASE

A STOP TASK is a single OPTION CARD with the single option STOP (and, of course, the card terminator 1111). Thus:

- STOP __ 1111

A DATA CASE may be a variety of things, depending on the input options requested. We shall examine the forms for all input options.

If the input option is IRAW or IPCT, you need a TITLE CARD, a TP CARD to give (in this order) the number of types and the number of sites, a SITE NAME CARD to give the names of the P sites, and a DATA ARRAY giving the raw counts or percentages. Thus:

- TITLE CARD
- TP CARD
- SITE NAME CARD
- DATA ARRAY

If the input option is ICOR, you need almost the same thing as for IRAW or IPCT. The differences are that the number of types is irrelevant (or unknown) so you use a P CARD instead of a TP CARD and the data array is assumed to be square, only one dimension is required. Thus:

- TITLE CARD
- P CARD
- SITE NAME CARD
- DATA ARRAY

If the input option is IADD, you also have one of the options IRAW, IPCT. The form of the DATA CASE follows the appropriate one of those two forms. IADD simply serves to prevent the loss of the information already in the machine. One caution: in order for the result of combining two data sets this way to be meaningful, it is essential that the number and order of types correspond exactly.

If the input option is IORD, the information required is the number of sites in the new trial ordering, given by a P CARD, and the new order
Figure C-1. Input form for a TASK which reads and orders raw data.
itself, given by a SITE NAME CARD. Thus:

\begin{verbatim}
P CARD
SITE NAME CARD
\end{verbatim}

If the input option is ILST, no further information is needed, so the DATA CASE is empty.

If the input option is IORG, no further information is needed, so the DATA CASE is empty.

If the input option is IGRP, you need to know how many combinations are to be made (use a P CARD) and what they are (via several COMBINING CARDS). Thus:

\begin{verbatim}
P CARD
COMBINING CARD
COMBINING CARD
 .
 .

COMBINING CARD
\end{verbatim}

Examples

Attached are a number of listings of card decks actually used under the PHOENIX system. They should provide sufficient examples for most users.

Figure C-1 shows a collection of raw data. The options request PHOENIX to read a set of raw data, perform a Permutation Search ordering, and print the raw data, the percentages, and the final correlation matrix. The second option card requests that only the correlation matrix be printed at the remote teletypewriter (TTY) station (the comment ‘TTY OPTIONS’ is completely ignored). The third line is simply a title. The fourth line specifies that the data set contains 57 types and 13 sites. The fifth line gives the names of the sites in the order of the data. The remainder of the data set is a table of raw counts of artifacts in sites. Note that it is not necessary to punch the numbers in columns; we do, however, find it convenient for proofreading.

Figure C-2 shows a collection of three sets of published data. This actually comprises three TASKs as defined above. The first asks that the data be read as percentages; the second two request that it be interpreted as precomputed correlation matrices. If the cards listed here were followed by the card STOP

\begin{verbatim}
they would then form the input for a complete run under PHOENIX.
\end{verbatim}

Figure C-3 shows the input for a complete run which performs several operations on a single data set. The first TASK calls for reading a table of percentages and printing the percentages and correlation matrix without
any application of ordering techniques. The second TASK calls for a Permutation Search on the input order; the third calls for the Type-Percent variant of Permutation Search. The fourth asks that a new starting order be set up and then used as a starting order for Permutation Search; the fifth does the same for the Type-Percentage technique. The sixth TASK rereads the input, computes a Presence-Absence matrix instead of the Brainerd-Robinson matrix which was inferred when the data were first read, and prints the correlation matrix for the input order. The seventh and final task calls for a Permutation Search on the newly-computed Presence-Absence matrix. The final line is the STOP TASK.
THREE TRENCHES -- FROM ROBINSON'S PAPER (1951)

8, 8,
IIA IIB IIC IA IB IIIA IIIB IIIC
24.0, 1.4, .2,11.3, .3,29.6,54.3, 0.0,
65.8, .9, 0.0, 4.5, 8.0, 0.0, 3.5, 0.0,
1.3, 0.0, .2, 3.8, .2,14.1,14.0, 6.6,
0.0, 0.0, 0.0, 1.3, .2, 0.0, 1.8, 3.3,
0.0, 0.0, 0.0, 3.3, .5, 0.0, 3.5, 2.5,
4.0, 0.0, 0.0, 24.9, 1.4, 7.0, 7.0, 27.5,
0.0, 0.9, 0.9, 3.5, 0.0, 5.4, 5.6, 57.1,
3.9, 0.0, .3, 2.8, 0.0, 49.3, 1.8, 0.0,

IPCT MXPM OCOR IIII
OCOR IIII
THREE TRENCHES -- FROM ROBINSON'S PAPER (1951)
8, 8,
IIA IIB IIC IA IB IIIA IIIB IIIC
24.0, 1.4, .2,11.3, .3,29.6,54.3, 0.0,
65.8, .9, 0.0, 4.5, 8.0, 0.0, 3.5, 0.0,
1.3, 0.0, .2, 3.8, .2,14.1,14.0, 6.6,
0.0, 0.0, 0.0, 1.3, .2, 0.0, 1.8, 3.3,
0.0, 0.0, 0.0, 3.3, .5, 0.0, 3.5, 2.5,
4.0, 0.0, 0.0, 24.9, 1.4, 7.0, 7.0, 27.5,
0.0, 0.9, 0.9, 3.5, 0.0, 5.4, 5.6, 57.1,
3.9, 0.0, .3, 2.8, 0.0, 49.3, 1.8, 0.0,

Figure C-3. Input for a complete PHOENIX run.
APPENDIX D

PROGRAM LISTING FOR PHOENIX-II
The PHOENIX Chronological Ordering System is written in ALGOL-20, a dialect of ALGOL-60 which runs on the CDC G-21 at the Carnegie Institute of Technology. There are five differences between ALGOL-20 as used in PHOENIX-II and pure ALGOL-60.

First, ALGOL-20 has a comment convention which we find more convenient than the ALGOL-60 comment statement. Any text to the right of a \( \texttt{;} \) on a card image is in general ignored. The exception is an occurrence within a string. We have taken advantage of this to set off comments to the right of the pertinent statements and to replace input-output and other machine-dependent commands by comments describing the operations to be performed. We feel that this will be more meaningful to anyone trying to understand or copy the code than the original commands would be.

Second, ALGOL-20 contains a capability for operating with variables of type \texttt{logic}. As used here, these are simple character string variables; their use in the program is restricted to reading, printing, and table lookups on site names and options.

Third, ALGOL-20 permits the use of periods within identifiers. They are used purely for mnemonic value and readability; they are completely ignored by the translator. Thus, the following are all instances of the same identifier.

\[
\text{OLDNORM} \quad \text{OLD.NORM} \quad \text{O.L.D.N.O.R.M.}
\]

Fourth, the G-21 character set requires certain simple character substitutions:

\[
\begin{align*}
* & \text{ for } \times \\
\Rightarrow & \text{ for } > \\
\leftarrow & \text{ for } \neq \\
\rightarrow & \text{ for } < \\
\end{align*}
\]

Fifth, ALGOL-20 arithmetic expressions may contain the truncation operator "\( \downarrow \)" defined by

\[
\downarrow X = \text{sign}(X) \times \text{entier}(|X|).
\]

Finally, we should note that we do not take advantage of any recursive ability of ALGOL-60.

We use these additional facilities without hesitation, for we feel that they contribute substantially to ease in using the system and ease in understanding the code.
BEGIN
INTEGER P,
   TT;
P := 35;
TT := 601;
BEGIN
INTEGER I,
   J,
   DMRG.P,
   BASE.P,
   BENV,
   SELECT,
   XOP,
   PSI, PSJ, PSK,
   II, JJ, KK, LL,
   J, I, J, K;
INTEGER DMQR,
   DMXPY,
   D1AA,
   D1AC,
   D1AO,
   D1AD,
   D1AM,
   D1AQ,
   D1AX,
   D1AY,
   D1AZ,
   DMAT,
   D2AF,
   D2AG,
   D2AH,
   D2AI,
   D2AJ,
   D2AK,
   D2AL,
   D2AM,
   D2AN,
   D2AO,
   D2AP,
   D2AQ,
   D2AR,
   D2AS,
   D2AT,
   D2AU,
   D2AZ,
   D2BA,
   D2BB,
   D2BC,
   D2BD,
   D2BE,
   D2BF,
   D2BG,
   D2BH,
   D2BI,
   D2BJ,
   D2BK,
   D2BL,
   D2BM,
   D2BN,
   D2BO,
   D2BP,
   D2BQ,
   D2BR,
   D2BS,
   D2BT,
   D2BU,
   D2BV,
   D2BW,
   D2BX,
   D2BY,
   D2BZ,
   D2C0,
   D2C1,
   D2C2,
   D2C3,
   D2C4,
   D2C5,
   D2C6,
   D2C7,
   D2C8,
   D2C9,
   D2CA,
   D2CB,
   D2CC,
   D2CD,
   D2CE,
   D2CF,
   D2CG,
   D2CH,
   D2CI,
   D2CJ,
   D2CK,
   D2CL,
   D2CM,
   D2CN,
   D2CO,
   D2CP,
   D2CQ,
   D2CR,
   D2CS,
   D2CT,
   D2CU,
   D2CV,
   D2CW,
   D2CZ,
   D2D0,
   D2D1,
   D2D2,
   D2D3,
   D2D4,
   D2D5,
   D2D6,
   D2D7,
   D2D8,
   D2D9,
   D2DA,
   D2DB,
   D2DC,
   D2DD,
   D2DE,
   D2DF,
   D2DG,
   D2DH,
   D2DI,
   D2DJ,
   D2DK,
   D2DL,
   D2DM,
   D2DN,
   D2DO,
   D2DP,
   D2DQ,
   D2DR,
   D2DS,
   D2DT,
   D2DU,
   D2DV,
   D2DW,
   D2DX,
   D2DY,
   D2DZ,
   D2E0,
   D2E1,
   D2E2,
   D2E3,
   D2E4,
   D2E5,
   D2E6,
   D2E7,
   D2E8,
   D2E9,
   D2EA,
   D2EB,
   D2EC,
   D2ED,
   D2EE,
   D2EF,
   D2EG,
   D2EH,
   D2EI,
   D2EJ,
   D2EK,
   D2EL,
   D2EM,
   D2EN,
   D2EO,
   D2EP,
   D2EQ,
   D2ER,
   D2ES,
   D2ET,
   D2EU,
   D2EV,
   D2EW,
   D2EX,
   D2EY,
   D2EZ,
   D2F0,
   D2F1,
   D2F2,
   D2F3,
   D2F4,
   D2F5,
   D2F6,
   D2F7,
   D2F8,
   D2F9,
   D2FA,
   D2FB,
   D2FC,
   D2FD,
   D2FE,
   D2FF,
   D2FG,
   D2FH,
   D2FI,
   D2FJ,
   D2FK,
   D2FL,
   D2FM,
   D2FN,
   D2FO,
   D2FP,
   D2FQ,
   D2FR,
   D2FS,
   D2FT,
   D2FU,
   D2FW,
   D2FX,
   D2FY,
   D2FZ,
   D2G0,
   D2G1,
   D2G2,
   D2G3,
   D2G4,
   D2G5,
   D2G6,
   D2G7,
   D2G8,
   D2G9,
   D2GA,
   D2GB,
   D2GC,
   D2GD,
   D2GE,
   D2GF,
   D2GG,
   D2GH,
   D2GI,
   D2GJ,
   D2GK,
   D2GL,
   D2GM,
   D2GN,
   D2GO,
   D2GP,
   D2GQ,
   D2GR,
   D2GS,
   D2GT,
   D2GU,
   D2GV,
   D2GW,
   D2GX,
   D2GY,
   D2GZ,
   D2H0,
   D2H1,
   D2H2,
   D2H3,
   D2H4,
   D2H5,
   D2H6,
   D2H7,
   D2H8,
   D2H9,
   D2HA,
   D2HB,
   D2HC,
   D2HD,
   D2HE,
   D2HF,
   D2HG,
   D2HH,
   D2HI,
   D2HJ,
   D2HK,
   D2HL,
   D2HM,
   D2HN,
   D2HO,
   D2HP,
   D2HQ,
   D2HR,
   D2HS,
   D2HT,
   D2HU,
   D2HV,
   D2HW,
   D2HX,
   D2HY,
   D2HZ,
   D2I0,
   D2I1,
   D2I2,
   D2I3,
   D2I4,
   D2I5,
   D2I6,
   D2I7,
   D2I8,
   D2I9,
   D2IA,
   D2IB,
   D2IC,
   D2ID,
   D2IE,
   D2IF,
   D2IG,
   D2IH,
   D2II,
   D2IJ,
   D2IK,
   D2IL,
   D2IM,
   D2IN,
   D2IO,
   D2IP,
   D2IQ,
   D2IR,
   D2IS,
   D2IT,
   D2IU,
   D2IV,
   D2IW,
   D2IX,
   D2IY,
   D2IZ,
   D2J0,
   D2J1,
   D2J2,
   D2J3,
   D2J4,
   D2J5,
   D2J6,
   D2J7,
   D2J8,
   D2J9,
   D2JA,
   D2JB,
   D2JC,
   D2JD,
   D2JE,
   D2JF,
   D2JG,
   D2JH,
   D2JI,
   D2JJ,
   D2JK,
   D2JL,
   D2JM,
   D2JN,
   D2JO,
   D2JP,
   D2JQ,
   D2JR,
   D2JS,
   D2JT,
   D2JU,
   D2JV,
   D2JW,
   D2JX,
   D2JY,
   D2JZ,
   D2K0,
   D2K1,
   D2K2,
   D2K3,
   D2K4,
   D2K5,
   D2K6,
   D2K7,
   D2K8,
   D2K9,
   D2KA,
   D2KB,
   D2KC,
   D2KD,
   D2KE,
   D2KF,
   D2KG,
   D2KH,
   D2KI,
   D2KJ,
   D2KK,
   D2KL,
   D2KM,
   D2KN,
   D2KO,
   D2KP,
   D2KQ,
   D2KR,
   D2KS,
   D2KT,
   D2KU,
   D2KV,
   D2KW,
   D2KX,
   D2KY,
   D2KZ,
   D2L0,
   D2L1,
   D2L2,
   D2L3,
   D2L4,
   D2L5,
   D2L6,
   D2L7,
   D2L8,
   D2L9,
   D2LA,
   D2LB,
   D2LC,
   D2LD,
   D2LE,
   D2LF,
   D2LG,
   D2LH,
   D2LI,
   D2LJ,
   D2LK,
   D2LL,
   D2LM,
   D2LN,
   D2LO,
   D2LP,
   D2LQ,
   D2LR,
   D2LS,
   D2LT,
   D2LU,
   D2LV,
   D2LW,
   D2LX,
   D2LY,
   D2LZ,
   D2M0,
   D2M1,
   D2M2,
   D2M3,
   D2M4,
   D2M5,
   D2M6,
   D2M7,
   D2M8,
   D2M9,
   D2MA,
   D2MB,
   D2MC,
   D2MD,
   D2ME,
   D2MF,
   D2MG,
   D2MH,
   D2MI,
   D2MJ,
   D2MK,
   D2ML,
   D2MM,
   D2MN,
   D2MO,
   D2MP,
   D2MQ,
   D2MR,
   D2MS,
   D2MT,
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION 139

BOOLEAN ARRAY
AS. RAW,
QUERY

REAL ARRAY
RAW(11ITT,11IPPI),
A(11ITT,11IPPI),
X(11IPPI,11IPPI),
Z(11IPPI),
SEEK(11ITT),
TRIAL(11IPPI),
BEST(11IPPI),
SH(11ITT),
PLT(1115),
IDP(11IPPI),
IDT(11ITT),
TOPC(1111),

BOOLEAN ARRAY
OPTION(11501);

INTEGER ARRAY
TTY.OPT(11501);

LOGIC ARRAY
RAME.NAM(11IPPI),
RAME.LIST(11IPPI),
TITLE(11120),
OPT.WAT(11501),
R0.OPT(11141);

This routine evaluates the norm

REAL, PROCEDURE EVAL(AVX,ORDER);
REAL ARRAY ORDER;
REAL ARRAY AVX;

This routine takes the SJH-DF-ERRORS norm

BEGIN
IF OPTION(TOPC) THEN GO TO T1;

This section forms the correlation matrix

SJH := 0.0;
FOR J := 1 STEP 1 UNTIL P DO
BEGIN
W := 0.0;
FOR JJ := 1 STEP 1 UNTIL J-1 DO
BEGIN
J := ORDER(JJ);
K := ORDER(JJ+1);
V := A(R,JK) - A(R,JK);
IF V < 0 THEN W := W - V;
END JJ LOOP;
END J LOOP;
FOR JJ := J+1 STEP 1 UNTIL P DO
BEGIN
J := ORDER(JJ-1);
K := ORDER(JJ);
V := A(R,JK) - A(R,JK);
IF V < 0 THEN W := W - V;
END JJ LOOP;

This routine gives a norm for ORDER

'ORDER' OF ARRAY
'ARRAY' DIFFERENCES TO PSDF

Options

PROSBE ORIGIHAL INPUT FORM
ASSORTED SWITCH

RAW DATA
PERCENTAGES
CORRELATION MATRIX
PRESENT ORDER
HIGHEST PCT OF EACH TYPE
TEMP FOR PERM SCH
FOR DELAYED PERM SCH
ENVELOPE OF VALUES FOR TPCT
HEIGHTS OF COLUMNS FOR TPCT
WHICH PITS TO PLOT THIS PAGE
IDENTITY ORDER FOR PITS
IDENTITY ORDER FOR TYPES
TYPE-PCT DIFFERENCE TOTALS
REMAINDERS: PERM SCH

OPTION SELECTION VECTOR

TELETYPICAL OUTPUT CONTROL

ALGOL-20 CONSTRUCT
NAMES OF THE P SITES
INPUT BUFFER FOR SITE LISTS
TITLE OF CURRENT DATA CASE
CHARACTER STRINGS FOR OPTIONS
INPUT BUFFER FOR OPTION READ

This routine evaluates the norm

VALUATE(SJH,ORDER);
GIVE A NORM FOR ORDER
ORDER' OF ARRAY
ARRAY DIFFERENCES TO PSDF

This routine takes the SJH-DF-ERRORS norm

BEGIN
IF OPTION(TOPC) THEN GO TO T1;

This section forms the correlation matrix

SJH := 0.0;
FOR J := 1 STEP 1 UNTIL P DO
BEGIN
W := 0.0;
FOR JJ := 1 STEP 1 UNTIL J-1 DO
BEGIN
J := ORDER(JJ);
K := ORDER(JJ+1);
V := A(R,JK) - A(R,JK);
IF V < 0 THEN W := W - V;
END JJ LOOP;
END J LOOP;
FOR JJ := J+1 STEP 1 UNTIL P DO
BEGIN
J := ORDER(JJ-1);
K := ORDER(JJ);
V := A(R,JK) - A(R,JK);
IF V < 0 THEN W := W - V;
END JJ LOOP;

This routine gives a norm for ORDER

'ORDER' OF ARRAY
'ARRAY' DIFFERENCES TO PSDF

Options

PROSBE ORIGIHAL INPUT FORM
ASSORTED SWITCH

RAW DATA
PERCENTAGES
CORRELATION MATRIX
PRESENT ORDER
HIGHEST PCT OF EACH TYPE
TEMP FOR PERM SCH
FOR DELAYED PERM SCH
ENVELOPE OF VALUES FOR TPCT
HEIGHTS OF COLUMNS FOR TPCT
WHICH PITS TO PLOT THIS PAGE
IDENTITY ORDER FOR PITS
IDENTITY ORDER FOR TYPES
TYPE-PCT DIFFERENCE TOTALS
REMAINDERS: PERM SCH

OPTION SELECTION VECTOR

TELETYPICAL OUTPUT CONTROL

ALGOL-20 CONSTRUCT
NAMES OF THE P SITES
INPUT BUFFER FOR SITE LISTS
TITLE OF CURRENT DATA CASE
CHARACTER STRINGS FOR OPTIONS
INPUT BUFFER FOR OPTION READ
RICE UNIVERSITY STUDIES

III THE FOLLOWING ARE OUTPUT ROUTINES. THE CODE FOR SOME
III MACHINE-DEPENDENT PROCEDURES HAS BEEN REPLACED BY
III PROSE DESCRIPTIONS.

PROCEDURE LINE;
   /// USPACE PRINTED PAPER BY ONE LINE.

PROCEDURE PAGE;
   /// USPACE PRINTER PAPER TO TOP OF NEXT PAGE.

PROCEDURE VECT.OUT(ARR, IDC, C);
   REAL ARRAY ARR;
   REAL ARRAY IDC;
   INTEGER C;

   /// PRINT THE (DECIMAL) ELEMENTS OF THE VECTOR ARR IN THE ORDER
   /// GIVEN BY THE VECTOR IDC. THERE ARE C PERTINENT ELEMENTS.
   /// IF OPTION.DOUBLE. is true then DOUBLE-SPACE THE PRINTING.
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION

PROCEDURE ARR.OUT(ARR, IDR, IDC, CI)
REAL ARR, IDR, IDC
INTEGER C

PRINT THE (DECIMAL) ELEMENTS OF THE TWO-DIMENSIONAL ARRAY
ARR, WITH ROWS IN THE ORDER GIVEN BY VECTOR IDR AND COLUMNS
IN THE ORDER GIVEN BY THE VECTOR IDC: THERE ARE R
PERTINENT ROWS AND C PERTINENT COLUMNS. IF OPTION=DCL
IS TRUE THEN DOUBLE-SPACE THE PRINTING.

PROCEDURE NAH.OUT(ARR, IDC, CI)
LOGIC ARRAY ARR
REAL ARRAY IDC
INTEGER C

PRINT THE (SYMBOLIC) ELEMENTS OF THE VECTOR ARR IN THE ORDER
GIVEN BY THE VECTOR IDC: THERE ARE C PERTINENT ELEMENTS.

PROCEDURE PR.TITLE
PRINT THE CARD IMAGE SAVED AS THE DATA CASE TITLE.

PROCEDURE PR.TYP(PAR)
INTEGER PAR

IF PAR = 1, THEN ON PRINTING AT THE REMOTE STATION.
IF PAR = 0, THEN OFF PRINTING AT THE REMOTE STATION.
THE ROUTINE HAS NO EFFECT.

PROCEDURE HEADING.PRINT
PRINT THE LEAD PAGE WHEN THE SYSTEM IS EXECUTED.
THE CONTENT OF THIS PAGE MAY VARY ACCORDING TO
THE TASTE OF THE INDIVIDUAL SYSTEM OR PROGRAMMER.
IN ADDITION, INITIALIZATION TASKS MAY BE INCLUDED
IN THIS PROCEDURE.

PROCEDURE MESS.OUT(SELECT)
INTEGER SELECT

PRINT THE MESSAGE SELECTED BY THE PARAMETER SELECT:
1. 'RAW DATA'
2. 'PERCENTAGES'
3. 'CORRELATION MATRIX'
4. 'GAME ENDS IN A TIE'
5. 'NEW CASE'
6. 'NEW ORDER FOR LAST INPUT'
7. 'USE LAST ORDER FOR LAST INPUT'
8. 'REVERT TO ORIGINAL INPUT ORDER'
9. 'COMBINE SITES AS REQUESTED'
ANY OTHER INPUT GIVES ERROR MESSAGE. THE MESSAGE
PRINTED IS PRECEDED AND FOLLOWED BY ONE EXTRA LINE.
PROCEDURE REPORT(SEL);<PAR1>,<PAR2>,<PAR3>)
INTEGER SEL;
INTEGER PAR1, PAR2);
REAL PAR3)
III PRINT THE MESSAGE WHOSE NUMBER IS GIVEN BY SELECT.
III PAR1, PAR2, AND PAR3 ARE VALUES TO BE PRINTED IN THE
III MESSAGES AN INDICATED BY <PAR1>, <PAR2>, AND <PAR3>.
III
1. 'NORM IS <PAR3> FOR ORDER, COLUMN ERRORS, AND
INTER-COLUMN ERRORS.'
III THIS MESSAGE IS FOLLOWED BY CALLS ON ROUTINES NAM.OUT
III AND VECT.OUT TO PRINT THE SITE NAMES AND APPROPRIATE
III ERROR VECTOR.
III
2. 'INTERCHANGE <PAR1> WITH <PAR2>; NORM IS NOW <PAR3>.'
III THIS MESSAGE IS FOLLOWED BY CALLS ON ROUTINES NAM.OUT
III AND VECT.OUT TO PRINT THE SITE NAMES AND APPROPRIATE
III ERROR VECTOR.
III
3. 'PUT <PAR2> IN POSITION <PAR2>; NORM IS NOW <PAR3>.'
III THIS MESSAGE IS FOLLOWED BY A CALL ON NAM.OUT TO
III PRINT THE SITE NAMES IN THE NEW ORDER.

PROCEDURE PR.RAW;
BEGIN
MESS.DU(1)
NAM.OUT(SITE, NAMES, L,P)'
LINES
ARR.OUT(RAW, IDT, T, L,P)'
LINES
END PROCEDURE PR.RAW;

I PRINT THE RAW DATA
I RAW DATA

PROCEDURE PR.PCT;
BEGIN
MESS.DU(2)
NAM.OUT(SITE, NAMES, L,P)'
LINES
ARR.OUT(A, IDT, T, L,P)'
LINES
END PROCEDURE PR.PCT;

I PRINT THE PERCENTAGES
I PERCENTAGES

PROCEDURE PR.CORR;
BEGIN
MESS.DU(3)
NAM.OUT(SITE, NAMES, L,P)'
LINES
ARR.OUT(X, L,P, L,P)'
LINES
REPORT((X,L,P), EVAL(X,L,P))
LINES
END PROCEDURE PR.CORR;

I PRINT THE CORRELATION MATRIX
I CORRELATION MATRIX

I NORM AND THAT STUFF
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION 143

PROCEDURE AXES(I1)
INTEGERS I1, I2
FOR I1 = 0 STEP 1 UNTIL I1-1 DO
BEGIN
I PUT A '1' IN EVERY EIGHTH POSITION STARTING WITH THE
FOURTH AND RUNNING FOR I1 PLACES.
END LOOP FOR SETTING UP VERTICAL AXES1
END PROCEDURE AXES;

PROCEDURE BAND;
BEGIN
AXES(KK-JJ+1)
RESET CHARACTER POINTER TO COLUMN 4
FOR I = 1 STEP 1 UNTIL 8*(KK-JJ)+4 DO
IF FISH[I] THEN BEGIN
SET UP A DENSE CHARACTER
ELSE BEGIN
SET CHARACTER POINTER AHEAD A SPACE
END CONDITIONAL WHICH SENDS POOL TO LINE;
PRINT LINE JUST SET UP AND JSPACE
END PROCEDURE BAND;

PROCEDURE SWATH;
BEGIN
FOR K1 = 1 STEP 1 UNTIL T DO
BEGIN
AXES(KK-JJ+1)
PRINT WHAT HAS BEEN SET UP AND ADVANCE PAPER
FOR J = 0 STEP 1 UNTIL K-JJ DO
END LOOP;
FOR J = 4 STEP 1 UNTIL 8*(K-JJ)+4 DO
BEGIN
Z[I] = FISH[J]
Y[I] = FISH[J+4]-Z[I]
END LOOP;
FOR I = 1 STEP 1 UNTIL 7 DO
FISH[J+1] = I*(Y7.8)+Z[I]
END J LOOP;
LL[I] = ![A(K,PEAK[KI])]/4.4+.9999)
FOR J = 1 STEP 1 UNTIL 8*(K-JJ)+4 DO
FISH[I] = !(FISH[J]/4.4+.9999))
FOR J = LL RTFP -1 UNTIL 2 DO BAND;
J = 31
SET UP THE TYPE NUMBER (ROW NUMBER) IN COLUMNS
ONE TO THREE OF THE PRINT LINE
BAND;
FOR J = 2 STEP 1 UNTIL LL DO BAND;
END J LOOP
AXES(KK-JJ+1)
PRINT WHAT HAS BEEN SET UP AND ADVANCE PAPER
PAGE1
END PROCEDURE SWATH;
PROCEDURE BATTLESHIP(I)
REAL ARRAY L;
BEGIN
IF I = 0 THEN
BEGIN
ERROR MESSAGE;
I CAN'T PLOT A TYPE=PERCENTAGE TABLE WITHOUT PERCENTAGES
GO TO EXIT;
END I = 0 (CONDITIONAL);
END
FOR II = 1 STEP 1 WHILE 15*(II-1) < P DO
BEGIN
PAGE;
LINE;
JJ = 14*II-1;
KK = JJ+14;
IF KK > P THEN KK = P;
PRINT HEADER FOR THIS SWATH OF 15 OR FEWIER SITES;
IT SHOULD GIVE SITE NUMBERS ON THIS SEGMENT AND THE
TITLE OF THE DATA CASE;
FOR I = JJ STEP 1 UNTIL (II+1):
PLOT I I-[JJ+11]:=(II+1)
NAME I OUT (SITE, NAME, PLOT, L, KK-JJ+11);
LINE;
SWATH:
END II LOOP;
END PROCEDURE BATTLESHIP;

III CONTOUR PLOTTING ROUTINE

PROCEDURE CONTOUR(I, ID, JD, IDCHARS, IPTS, JLINES, JPTS, FINTRY, LBL, OPTION);
VALUE ID, JD, IDCHARS, IPTS, JLINES, JPTS, FINTRY, OPTION;
REAL ARRAY FI
REAL FINTRY;
INTEGER ID, JD, IDCHARS, JLINES, JPTS, OPTION;
LOGIC ARRAY LBL;
BEGIN
HALF STPWHL, STPWHL, WHLGRD, STP_VD, STPLN, RLSIDE, FJLNL, FLOTJ, DJTJ,
HDTJ, FDHE, FDHE, FKL, GD, FDV, FSW, PMAXX, PMIN, CP, PMIN, PRAX, GL, P, PAGEJ;
INTEGER KSTAP, KLNI, K, ISWTCH, JSWITCH, LSIDE, NDCHAR, ITCHARS, NPAGES,
NCARRS, WKLNL, IFF, IMJ, IMK, JDF, JCEP, JCEP, JN, JNUM, NEY, IM, JAY, FKL, JGDL, IDDT,
KSTAP, KLIN, KSWTCH, JSWITCH, LSEI, NDCHAR, ITCHARS, NPAGES, NCARRS, WKLNL, IFF, IMJ, IMK, JDF, JCEP, JCEP, JN, JNUM, NEY, IM, JAY, FKL, JGDL, IDDT;
LOGIC ARRAY KPRTN(1130), LINES(1130);;
BEGIN
IF IDCHARS(IPTS) THEN GO TO 99991;
JLINES(IPTS) THEN GO TO 99991;
ISWTCH-OPTION MOD 2 + 11
ISWTCH-OPTION MOD 2 + 11
LSIDE = 119;
NDCHAR = 119;
ITCHARS(I, I, ) = (IDCHARS(IPTS));
NPAGES = ITCHARS(IPTS) + 1;
NCARRS = ITCHARS(IPTS);=
IF NPAGES MOD 1 = 0 THEN 33 TO L1002;
NPAGES = NPAGES / 33;
NCARRS = ITCHARS MOD 119;
L1002: NLMAIN1 = JD-1*Y(JLINES/IPTS)-11
ITPP = NPAGES * JPSTWHL/STPWHL+STPWHL/IPTS; WHLGRD = 1/STPWHL;
STPLN = JLINES; STPLN = STPLN*KPTS; KPRNT((IPTS)-1);
TRIPLE LOOP - START OF PLOTTING COMPUTATIONS

FIRST LOOP FOR KSTRP=1 STEP 1 UNTIL NAGES DO
BEGIN
PAGE
IF [PPS] THEN BEGIN
PAGE
END
PRINT VALUE OF KSTRP AND A LEGEND END
PR-TITLE: LSIDE=LSIDE+NCAR; FLSIDE=FLSIDE; IF KSTRP[PPS] THEN 50 TO L1004;
L1003NCAR=NCAR+2;
L1004KRTS=1;LSIDE=LSIDE+NCAR+1;
IMIN=(FLSIDE*STPLH+STPWH-1); (MAX[MIN([KRTSIDE*STPLH+STPWH-1.ID-1]])
FLIN=1.0001-1STPLV;
J=0
SECOND LOOP FOR KLV=1 STEP 1 UNTIL NKLH DO
BEGIN
I=J
FLIN+FLIN=STPL1
J=FLIN+FLIN+J+J-1
DJSTOP=FULL=FULL+STPLH+STPLH)
IDJOF-ID*ID-J)
JCOF-ID+ID-J)
FNE=FL1*JCOEF+DJSTOP+FULL*J, JCOEF*DJSTOP)
FSE=FNE*STPL1*FULL*J, JCOEF*DJSTOP)
FOR I=1 STEP 1 UNTIL N-1 DO PRINT[I+ I+1]
FKL=VNLHGRD *[FNL-2]+1.9]
THIRD LOOP FOR I=1 STEP 1 UNTIL I=MIN DO
BEGIN
FKL+FKL=VNLHGRD)
GQ=FKL*31085
JGOTO
FNW>FSE;
JCOEF-ID*(J-1)+I-J)
JCOEF-ID+I-J)
FNE=FL1*JCOEF*DJSTOP+FULL*J, JCOEF*DJSTOP)
FSE=FNE*STPL1*FULL*J, JCOEF*DJSTOP)
NUM=MVJ"FMTTVY
IF SIGV(UJ)<0 THEN GO TO L521
LS+LS=NUM+I-1
L521C=FMTTV
IF FMAT>C THEN GO TO SWL17[SWITCH]1F******
GO TO 54.7I[SWITCH]1
L77#NUMBER=(ABS[C]) MOD 10)+327
GO TO [.791]
L777#IF NUM[OD 2 = 0 THEN GO TO L7791
L777#NUMBER-11
L78#HAV+MAX+MAX+MIN+11
CL C+FW;
IF (CL=[C-FNE]) THEN BEGIN
\=2/\FNE-FNW)
\=MAX[MIN[P1, P]]
\=MAX[P1, P])
IF I+I+A
END1
L783#CL+C-FSE;
IF (CL=[C-FSE]) THEN BEGIN
\=2/\FSE-FNW)
\=MAX[P1, P])
\=MAX[P1, P])
IF I+I+A
END1
L777#IF (C-FW)*(C-FSW)<0 THEN GO TO L707
[CL+C-FMIN] [HAV+I+I]
L761: IF IF MAX<2 THEN BEGIN
    IF (C*FNE)*(C+FSE)>0 THEN GO TO L710
    END
    L71KSM=KSM+P(IVSTPWHL+GO)
    KLGO=MAX/STPWHL+GO1
    IF (KS4L*(KCHAR-KSM))>0 THEN GO TO L1521
    L152I: IF (KLS*(KCHAR-KLG))>0 THEN GO TO SWL154(IOUT)
    L153: IOUT=IOUT+1
    L154GOTO 5,154(IOUT)
    L155KSM=MAX(1,KSM)
    KLG=MIX(KCHAR,KLG)
    L156 FOR K<-KSM STEP 1 UNTIL KLG DO KPRNT(K1-NUMBER)
    GO TO L157
    L1571: IF JD1=8 THEN GO TO L203
    IF KO>0 THEN KPRNT(KO)=**
    L201:END I IF FD TO L2021
    L202I: IF JD1=0 THEN GO TO L203
    IF JD1#0 THEN 33 TO L202
    L203:KPRNT(K1**KPRNT(INCHAR)**

III PRINT THE LINE IMAGE (119 CHARACTERS) STORED ONE
    CHARACTER PER WORD IN KPRNT:
    END IF 1ST LOOP
    GO RETURN
    L99991

III PRINT ERROR MESSAGE
1 NO CONTOUR MAP PRODUCED. INCORRECT SCALE SPECIFICATION.
III CHECK FOURTH THROUGH EIGHTH ARGUMENTS OF 'CONTOUR'
    PAGE
    END OF CONTOURING PROCEDURE

III THIS IS THE OUTPUT CONTROL PROCEDURE

PROCEDURE PRINT OUTPUT

BEGIN
    IF OPTION(ODOCN1 THEN BEGIN
    PRINT(TTY,OPTION(Document1):
    PR.CORR
    PRINT1
    END PRINT CORR CONDITIONAL

    PR.CORR END

    IF OPTION(ODOCN1 THEN BEGIN
    PRINT(TTY,OPTION(Document1):
    PR.RAW
    PRINT()
    END PRINT RRAW CONDITIONAL

    PR.RAW

    IF OPTION(ODOCN1 THEN BEGIN
    PRINT(TTY,OPTION(DOCN1):
    PR.PCT
    PRINT()
    END PRINT PCT CONDITIONAL

    PR.PCT

    IF OPTION(ODOCN1 THEN BEGIN
    REAL ARRAY REOUIX1,11)
    FOR III= 1 STEP 1 UNTIL P DO
        FOR JJ= 1 STEP 1 UNTIL P DD
            REOUIX1,11= X(111,111)
            PRINT()
            CONTOUR(REOUIX1,11)
            END OF CONTOURING PROCEDURE

1 TEMPORARY MATRIX FOR PLOTTIN
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION 147

IF OPTION[0:999] THEN
BEGIN
PRINT(A); BATTLESHIP(L); END BATTLESHIP CONDITIONAL;
END PRINT, OUTPUT;

III THE FOLLOWING ARE INPUT ROUTINES

III GET THE CONTROL OPTIONS

PROCEDURE GET_OPTIONS(SELECT); INTEGER SELECT;
BEGIN
QUERY := TRUE;
IF SELECT = 1 THEN FOR J := 1 STEP 1 UNTIL 50 DO OPTION(J) := FALSE;
IF SELECT = 2 THEN FOR J := 1 STEP 1 UNTIL 50 DO TTY.OPT(J) := 0;
GET_MORE_OPTIONS;
READ 15 4-CHARACTER OPTIONS SEPARATED BY SPACES OR COMMAS INTO RD.OPT[1] THROUGH RD.OPT[15];
NOTE: "$" $"." $"." IN RD.OPT[1] DENOTED END OF INPUT STREAM.
END;
FOR I := 1 STEP 1 UNTIL 16 DO BEGIN
IF RD.OPT[I] = $"TT$ THEN GO TO TRY.AGAIN;
IF RD.OPT[I] = $"STOP$ THEN GO TO GET_MORE_OPTIONS;
FOR J := 1 STEP 1 UNTIL 50 DO IF RD.OPT[J] = OPT.VAR(J) THEN BEGIN
IF SELECT = 1 THEN OPTION(J) := TRUE; IF SELECT = 2 THEN TTY.OPT(J) := 1;
GO TO TRY.AGAIN;
END J LOOP;
PRINT ERROR MESSAGE;
OPTION NAME NOT IN TABLE AND OFFENDING NAME QUERY := FALSE;
GO TO TRY.AGAIN;
TRY.AGAIN;
END I LOOP;
GO TO GET_MORE_OPTIONS;
HAVE.ALL;
AWAY;
END PROCEDURE GET_OPTIONS;

III GENERAL INPUT SUPPORT ROUTINES

PROCEDURE G.TITLE;
READ THE NEXT CARD IMAGE IN THE INPUT STREAM AS A TITLE;
TITLE CARD FOR THIS DATA CASE,
PROCEDURE SITES(P):
    INTEGER P;
    READ P 4-CHARACTER SITE NAMES INTO VECTOR SITE.NAM.;

PROCEDURE ISENT(P,L)
    ARRAY L;
    INTEGER P;
    FOR I := 1 STEP 1 UNTIL P DO
        L[I] := I;

PROCEDURE TLUI(LOOK,FIND)
    INTEGER FIND;
    LOGIC LOOK;
    BEGIN
        INTEGER I;
        QUERY := TRUE;
        FOR I := 1 STEP 1 UNTIL ORI.L DO
            IF SITE.NAM(I) = LOOK THEN
                BEGIN
                    FIND := I;
                    GO TO TUI;
                END;
                QUERY := FALSE;
                FIND := 0;
            END:
            TUI:
            END PROCEDURE TLUI;

PROCEDURE COMP.DAJ
    BEGIN
        FOR J := 1 STEP 1 UNTIL P DO
            FOR I := J STEP 1 UNTIL P DO
                BEGIN
                    JJ := L[I];
                    KK := L[J];
                    W := 0.0;
                    FOR I := 1 STEP 1 UNTIL 1 DO
                        FOR J := 1 STEP 1 UNTIL P DO
                        END;
                    END;
                END:
            END:
        END PROCEDURE COMP.DAJ;

PROCEDURE COMP.PM;
    BEGIN
        FOR J := 1 STEP 1 UNTIL P DO
            FOR I := J STEP 1 UNTIL P DO
                BEGIN
                    W := 0.0;
                    FOR I := 1 STEP 1 UNTIL 1 DO
                        IF (A[I,JJ] = (I-1)) + (I,JJ = I) THEN
                            V := W + 1;
                        END;
                    END;
                END:
            END:
        END PROCEDURE COMP.PM;
PROCEDURE COMP.PCT;
    BEGIN
        FOR J := 1 STEP 1 UNTIL P DO
        BEGIN
            SUM := 0;
            FOR I := 1 STEP 1 UNTIL T DO
                SUM := SUM + RAW[I,J];
            IF SUM > 0 THEN
                GO TO MULTI;
            FOR I := 1 STEP 1 UNTIL T DO
                A[I,J] := 100.0 * RAW[I,J] / SUM;
        END J LOOP;
    END PROCEDURE COMP.PCT;

PROCEDURE COMP.PEAK;
    BEGIN
        FOR J := 1 STEP 1 UNTIL T DO
        BEGIN
            W := 0;
            FOR I := 1 STEP 1 UNTIL P DO
                    BEGIN
                        W := A[I,J];
                        K := I;
                    END CONDITIONAL;
            PEAK[I,J] := K;
        END J LOOP;
    END PROCEDURE COMP.PEAK;

III MAIN INPUT CONTROL

PROCEDURE RRAW;
    BEGIN
        READ THE INTEGERS T AND P;
        SITES[P]:
        IF OPTION[RAW] THEN
            BEGIN
                READ T ROWS OF P ELEMENTS INTO ARRAY RAW
                P := P + RARE.P;
                COMP.PCT;
                COMP.PEAK;
                WAS.RAW := TRUE;
                END CONDITIONAL;
        IF OPTION[OPTION][PCT] THEN
            BEGIN
                FOR I := 1 STEP 1 UNTIL T DO
                    FOR J := 1 STEP 1 UNTIL P DO
                        RAW[I,J] := 0;
                III READ T ROWS OF P ELEMENTS INTO ARRAY A
                P := P + RARE.P;
                COMP.PEAK;
                WAS.RAW := FALSE;
            END CONDITIONAL;
    END PROCEDURE RRAW;

I COMPUTE PERCENTAGES
I LOAD VECTOR WITH PEAKS IN PCT
I FIND GREATEST PCT PER TYPE
I READ NEW DATA SET, EITHER RAW I OR PERCENTAGES
PROCEDURE R_COEFF;
BEGIN
  T := 1;
  P := 1 + BASE.P;
  SITES := P ELEMENTS INTO ARRAY X
  P := 1 + BASE.P;
  WAS_RAW := FALSE;
END PROCEDURE R_COEFF;

PROCEDURE REORDER;
BEGIN
  P := 1 + BASE.P;
  SITE_LIST[I] := SITE_NAME[I];
  FOR I := 1 STEP 1 UNTIL P DO
  BEGIN
    ZZ := SITE_LIST[I];
    SITE_LIST[I] := SITE_NAME[I];
    SITE_NAME[I] := ZZ;
  END; LOOP;
  FOR I := 1 STEP 1 UNTIL P DO
  BEGIN
    TLU(SITE_LIST[I], J);
    IF NOT THEN BEGIN
      PRINT ERROR MESSAGE;
      SITE NAME NOT IN TABLE;
      AND OFFENDING NAME;
      GO TO L.DUTY;
      END RENT RTUENTE;
    END LOOP;
  END LOOP;
END PROCEDURE REORDER;

PROCEDURE GROUP;
BEGIN
  P := 1 + BASE.P;
  SITE_NAME[I] := NAME TO
  IF WAS_RAW THEN
    FOR K := 1 STEP 1 UNTIL T DO
    RANK, BASE.P+1 := 0;
  ELSE
    FOR K := 1 STEP 1 UNTIL T DO
    A[K, BASE.P+1] := 0;
    FOR J := 1 STEP 1 UNTIL J DO
    BEGIN
      READ NEXT SITE NAME FROM SAME CARD. IT GOES TO WW
    TLU(WW, I);
    IF NOT THEN BEGIN
      PRINT ERROR MESSAGE;
      SITE NAME NOT IN TABLE;
      AND OFFENDING NAME;
      GO TO L.DUTY;
      END RENT RTUENTE;
    END;
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION

151

IF WAS.RAU THEN
FOR K = 1 STEP 1 JNTIL 1 DO
RAW(K,BASE,P+1] = RAW(K,BASE,P+1] + RAW(K,II]
ELSE
FOR K = 1 STEP 1 JNTIL 1 DO
AIK, BASE, P+1] = AIK, BASE, P+1] + AIK, III]
END JJ LOOP;
END II LOOP;
ORIG,P = ORIG,P + 1;
BASE,P = ORIG,P;
P = ORIG,P;
IF OPTION0IRAWI THEN COMP,?CTI
COMP.PGK;
END PROCEDURE GROUP;

PROCEDURE READ.INPUT
BEGIN
IF OPTION0ILSTI THEN IDENT(4,P)]
IF OPTION0IRAWI=OPTION0IPCTI THEN
BEGIN
PAGE;
G.TITLE;
IF OPTION0IADD] THEN BASE,P = ORIG.P
ELSE BASE,P = 0;
RAW;
IF OPTION0IBRI] THEN COMP,BR
ELSE IF OPTION0IBDM+PI THEN COMP,PM
ELSE COMP,PR;
MESS.OUT[5];
PR,TITLE;
END CONDITIONAL;
IF OPTION0ICORI THEN
BEGIN
PAGE;
G.TITLE;
BASE,P = 0;
R.CORP;
MESS.OUT(9);
PR,TITLE;
END CONDITIONAL;
IF OPTION0IORDI THEN
BEGIN
PAGE;
P = ORIG,P;
BASE,P = 0;
REORD;
MESS.OUT(9);
PR,TITLE;
NAM.OUT(SITF,NAM,L,P);
END CONDITIONAL;
IF OPTION0IORGI THEN
BEGIN
PAGE;
P = ORIG,P;
BASE,P = 0;
REORD;
MESS.OUT(9);
PR,TITLE;
NAM.OUT(SITF,NAM,L,P);
END CONDITIONAL;
IF OPTION0ILSTI THEN
BEGIN
MESS.OUT(73);
PR,TITLE;
NAM.OUT(SITF,NAM,L,P);
END CONDITIONAL;

I CONTROL FOR DATA READ

I WHETHER TO
I REPLACE
I OR ADD
I READ DATA AND PIT NAMES
I COMPUTE CORRELATION MATRIX
I NEW CASE

I NEW CASE

I LAST DATA, NEW ORDER

I LAST DATA, LAST ORDER
IF OPTIONIQGRPI THEN
BEGIN
PAGEI;
P := ORIG.P;
BASE.P := P;
MESS.OUT(q);
PR.TITLE;
GROUP;
NAM.OUT(SITE,NAN.L,P);
IF OPTIONIQVARI THEN COMP.BR
ELSE IF OPTIONIQMxrI THEN COMP.PM
ELSE COMP.PRI
END CONDITIONAL

END PROCEDURE READ.INPUT;

III HERE BEGINS THE SECTION FOR PROGRAMS FOR SERIATION TECHNIQUES

III FIRST PROCEDURE: PERMUTATION SEARCH

III PERMUTATION SEARCH: SEARCH PATTERNS

PROCEDURE NEXT(OLD,NEW,SELECT);
REAL ARRAY OLD;
REAL ARRAY NEW;
INTEGER SELECT;
BEGIN
IF SELECT = 0 THEN
BEGIN
PSI := 1
PSJ := 2
QUERY := TRUE;
GO TO EXIT;
END SELECT = 1

IF SELECT = 1 THEN
BEGIN
QUERY := TRUE;
FOR I = 1 STEP 1 UNTIL P DO
NEW(I) := OLD(I);  // 1ST SCHEME FOR NEXT ORDER
Z := OLD(PSI);
NEW(PSI) := OLD(PSJ);
NEW(PSJ) := Z;
IF OPTIONIQPSALI THEN REPORT(2,PSI,PSJ,EVAL(X,NEW));
IF PSJ < P THEN
PSJ := PSJ + 1;
ELSE IF PSJ+1 < P THEN
BEGIN
PSI := PSJ+1
PSJ := PSJ + 1
END CONDITIONAL SEGMENT
ELSE QUERY := FALSE;
END SELECT := 1

EXIT;
END PROCEDURE NEXT;
COMPUTER ANALYSIS OF CHRONOLOGICAL SERIATION

PROCEDURE NEXT2(OLD, NEW; SELECT);
REAL ARRAY OLD;
REAL ARRAY NEW;
INTEGER SELECT;
BEGIN
  IF SELECT = 0 THEN
    BEGIN
      PSI := 1;
      PSJ := 1;
      QUERY := TRUE;
      GO TO EXIT;
    END;
  END;
  IF SELECT = 1 THEN
  BEGIN
    QUERY := TRUE;
    FOR I := 1 STEP 1 UNTIL \( \text{INV(PSI, PSJ)} \) DO
      IF PSI > PSJ THEN
        BEGIN
          NEW[I] := OLD[I];
        END;
      IF PSI < PSJ THEN
        BEGIN
          FOR I := PSI+1 STEP 1 UNTIL PSJ DO
            NEW[I] := OLD[I-1];
          END;
        END;
    IF PSI = PSJ THEN
      BEGIN
        IF Query THEN GO TO EXIT;
      END;
    EXIT:
  END;
END PROCEDURE NEXT2;

III. PERMUTATION SEARCH: THE MAIN Routines

PROCEDURE PSCH.IKK;
BEGIN
  NEXT1(TRYAL.L, 0);
  TRIAL.NORM := EVAL(I, L);
  LOOP:
  BEGIN
    NEXT1(TRYAL.L, 1);
    TRIAL.NORM := EVAL(I, TRYAL.L);
    NOW.NORM := EVAL(I, TRYAL.L);
    IF NOW.NORM < TRIAL.NORM THEN
      BEGIN
        FOR I := 1 STEP 1 UNTIL P DO
          L[I] := TRYAL.L[I];
        TRIAL.NORM := NOW.NORM;
        IF OPTION(OPSTRI) THEN REPORT(3, PSI, PSJ, EVAL(I, NEW));
      END;
  END; END PROCEDURE PSCH.IKK;
PROCEDURE PSCH.DLY;
BEGIN
NEXT(L,TRIAL.L,0);
TRIAL.NORM := EVAL(X,L);
FOR I := 1 STEP 1 UNTIL P DO
BEST.L[I] := L[I];
END;
LOOP;
NEXT(L,TRIAL.L,1);
NOW.NORM := EVAL(X,TRIAL.L);
END:
IF NOW.NORM < TRIAL.NORM THEN BEGIN
FOR I := 1 STEP 1 UNTIL P DO
BEST.L[I] := TRIAL.L[I];
TRIAL.NORM := NOW.NORM;
END;
IF OPTION(OPTRI) THEN REPORT(4,0,0,NOW.NORM);
END:
IF QUERY THEN GO TO LOOP;
FOR I := 1 STEP 1 UNTIL P DO
L[I] := BEST.L[I];
END PSCH.DLY;
PROCEDURE PERMUTATION.SEARCH;
BEGIN
PRINT(TTY,OP1[PSCH]);
IF -OP1[OP1IM] = -OP1[OP2DLY] THEN BEGIN
OP1[OP1IM] := TRUE;
OP1[OP2DLY] := TRUE;
END;
OLDNORM := EVAL(X,L);
NOW.NORM := OLDNORM;
REPORT(1,0,0,OLDNORM);
FOR LOOP := 1 STEP 1 UNTIL 20 DO BEGIN
PRINT(TTY,OP1[OPSTRI]);
IF OP1[OP1IM] THEN PSCH.IMM;
IF OP1[OP2DLY] THEN PSCH.DLY;
PRINT(TTY,OP1[PSCH]);
NOW.NORM := EVAL(X,L);
REPORT(1,LOOP,0,NOW.NORM);
IF NOW.NORM = OLDNORM THEN GO TO THRU;
OLDNORM := NOW.NORM;
END LOOP LOOP;
MESS.OUT(4);1
THRU:
PRINT(0);
END PROCEDURE PERMUTATION.SEARCH;

III OTHER SERIATION PROCEDURES

PROCEDURE MEIGHAN;  
III THIS ROUTINE IS PROVIDED AS A FUTURE HOME FOR A PROGRAM  
III TO COMPUTE THE GENERALIZED MEIGHAN ORDERING FOR A SET  
III OF DATA.

PROCEDURE DEPSEV;  
III THE PROGRAM FOR CONTEXTUAL ANALYSIS IS NOT REPRODUCED  
III HERE.
PROGRAM FOR THE ASCHERI TECHNIQUE IS NOT REPRODUCED
HERE.

THE MAIN CONTROL ROUTINE OF THE PHENIX SYSTEM

FIRST WE SET UP ENTRIES IN OPTION VECTOR AND DEFINE
THE INDICES

IDENTITY ORDER FOR PITS
IDENTITY ORDER FOR TYPES
START WITH IDENTITY

END OF INITIATION. IF CODE IS TO BE DUMPED IN
BINARY TO AVOID RECOMPIILATION, DO IT HERE

CYCLE:
SET.OPTIONS(1)
IF QUERY THEN GO TO 1.JUIT
IF OPTION(2STOP) THEN GO TO 1.JUIT
SET.OPTIONS(2)
IF QUERY THEN GO TO 1.JUIT
IF OPTION(2STOP) THEN GO TO 1.JUIT
IF OPTION(4STOP) THEN GO TO 1.JUIT
IF OPTION(3STOP) THEN LEN = 7
ELSE LEN = 19
READ.INPUT
IF OPTION(JASCH) THEN ASCHERI
IF OPTION(JDPM) THEN DEMPSEY
IF OPTION(JEIGH) THEN EIGHAN
IF OPTION(JPSCH) THEN PERMUTATION-SEARCH
PRINT.OUTPUT
GO TO CYCLE
1.JUIT
PAGE
END
END ARRAY-SIZE CONTROL BLOCK
This appendix is directed toward programmers, particularly those who wish to set up versions of PHOENIX on some machine other than a CDC G-21. The first section discusses the organization of the system; the second is devoted to instructions for extending and altering the system.

Organization

It should be fairly straightforward to set up another version of PHOENIX. The major differences between ALGOL-20 and pure ALGOL-60 are discussed in Appendix D; techniques for converting pure ALGOL-60 to other local dialects of ALGOL have probably been developed by all groups using the dialects. As a supplement to the program listing of Appendix D, we provide here a set of notes on the individual procedures and their functional relationships.

The data structures are described in a general way in Appendix C and in detail by the declarations in Appendix D. The philosophy of the system is that it is more efficient to use key vectors to give the various permutations of the site order than it is to actually interchange data elements. Accordingly, the arrays RAW, A, and X are always left in their original orders and vectors L, TRIAL.L, BEST.L, PLOT.L, and IDP are used to give the permutations: If \( L[J] = K \), then the \( K^{th} \) site in the input order is the \( J^{th} \) site in the present order.

OUTER BLOCK:

The main body of PHOENIX is the inner block of the program. The outer block is used to control array sizes. The variables PP and TT represent the maximum number of sites (pits) and types. They may be adjusted to suit available memory or to accommodate particular extreme data sets. No processing is done in the outer block.

MAIN PROCESSING BLOCK:

DECLARATIONS:

Most of the variables are declared here. The ALGOL-20 comment convention is used to suggest the use of each variable or array.

PROCEDURE EVAL:

EVAL accepts as parameters the array to norm and the order to use on it. The norm to use is selected on the basis of the switch which chooses the normal or type-percentage option for Permutation Search.
OUTPUT SUPPORT PROCEDURES:

Procedures LINE, PAGE, VECT.OUT, ARR.OUT, NAM.OUT, PR. TITLE, PR.INT, HEADING.PRINT, MESS.OUT, REPORT are support routines for the main output procedures. They are also used by the input procedures and ordering techniques for messages and diagnostic output. The actual ALGOL-20 code for these procedures is highly machine-dependent. It has been replaced by comments describing the operations to be carried out.

Procedure PR.RAW: To print the raw data, call for the message ‘RAW DATA,’ print the P site names in the current order as given by the vector L, skip a line, print the raw data (RAW) with column order given by the identity-order vector (IDT) and row order given by vector L, and skip a line.

Procedures PR.PCT, PR.CORR: The procedures for printing the percentage table and the correlation matrix are similar to PR.RAW.

TYPE-PERCENTAGE GRAPH PLOTTER:

Procedure AXES sets up the vertical axes: a vertical bar (“|”) in every eighth column for as many columns as appropriate.

Procedure BAND sets up and prints one line of output. J gives the distance of this line from the center-line of the current type. FIXH gives the height of the plot for the current type for each print column. BAND sets up the densest character on the printer in each column for which \( \text{FIXH}[i] \geq J \) and leaves spaces or axis bars everywhere else. Note that the axis overrides a space and the plot character overrides the axis.

Procedure SWATH controls printing of the entire plot for as many sites as will fit on one page. To print the band for each type, it first moves the appropriate percentage values from array A to every eighth entry of vector FISH. It then performs a linear interpolation to obtain the other band widths (that is, to complete the envelope of the curve). The maximum width of the band (across the entire plot) is obtained by normalizing the largest percentage of the current type; this value is stored in LL. FIXH is then computed from FISH for use in BAND. Next, SWATH uses three for-loops to call BAND for each of the \((2\times LL-1)\) lines of output. The type number is printed with the contour line.

Procedure BATTLESHIP is the control procedure for type-percentage plots. Each type-percentage (or battleship) graph is plotted in as many vertical swaths of 15 columns as are necessary to plot all sites. The rightmost site of each swath is the leftmost site of the next in order to get interpolation in both directions. Each swath has a header consisting of the title of the data case and the site names positioned over the vertical axes.
CONTOUR PLOTTER:

Procedure CONTOUR: We wish to thank Ira Ruben for the use of his contour-plotting procedure.

MAIN OUTPUT CONTROL:

Procedure PRINT.OUTPUT is called from the main driver after all ordering procedures have been executed. It tests each of the terminal output options and prints the tables and plots requested.

INPUT SUPPORT PROCEDURES:

Procedure GET.OPTIONS is called twice by the driver. The first time (SELECT = 1), the options are recorded as ‘TRUE’ values in the Boolean vector OPTION. The second call (SELECT = 2) records options as 1’s in the integer vector TTY.OPT. Elements of OPTION are tested to control execution; elements of TTY.OPT are used as parameters to PRINT to control output to the remote station. The first section of the procedure initializes the proper vector to ‘FALSE’ or 0. Then cards are read into RD.OPT until ‘|||’ is encountered. Each 4-character option name read is looked up in the table OPT.NAM which has been initialized to the option names, and the parallel entry in OPTION or TTY.OPT is set to the value ‘TRUE.’ If an option name is not found in OPT.NAM, an error message is printed, the Boolean scalar QUERY is set FALSE, and the procedure returns to the driver, which then takes appropriate action.

Procedures G.TITLE, SITES: These are support routines for the main input procedures. The actual ALGOL-20 code is highly machine-dependent, and has been replaced by comments describing the operations to be carried out.

Procedure IDENT: Fill a vector with the identity permutation.

Procedure TLU finds entries in vector SITE.NAM and returns their positions as output. It is used by the procedures which reorder and group the sites (RE.ORD and GROUP).

Procedure COMP.BR computes a Brainerd-Robinson correlation matrix according to the formula given elsewhere.

Procedure COMP.PM computes a Presence-Absence (plus-and-minus) correlation matrix according to the formula given elsewhere.

Procedure COMP.PCT computes the percentage matrix from the raw data.

Procedure COMP.PEAK fills vector PEAK with the column numbers in which the maximum percentages for each of the types occur.

Procedure R.RAW reads both raw data and percentages. It begins filling the appropriate array at column BASE.P, and hence can be used both for reading completely new data sets and for augmenting old ones. The procedure first reads T and P; it then evaluates ORIG.P, which contains the active size of the arrays. It next reads the site names for the new
data, and clears RAW if percentages are being read. Then the procedure reads the actual values to be stored, reevaluates P to reflect the new values, computes percentages if necessary, executes COMP.PEAK, and sets the value of WAS.RAW to reflect the type of data read.

Procedure R.CORR reads the input data when it represents a precomputed correlation matrix. Its operation is similar to that of R.RAW.

Procedure RE.ORD reads a new order for the data and sets up the order in L. SITE.LIST is used as a temporary array in which to store the original order while reading the new order; TLU is used to find the original positions of the sites being reordered.

Procedure GROUP performs the operations necessary to combine sites. It reads the number of combinations to be made (H), and executes the combining loop H times. In the combining loop, a card is read to obtain the name of the new "site," the number of site counts to combine to get it, and the names of the sites being combined. The sum is then computed for each type of the sites being combined, the variables ORIG.P, BASE.P, and P are updated, COMP.PCT is executed if necessary, and COMP.PEAK is executed.

MAIN INPUT CONTROL:

Procedure READ.INPUT is called from the main driver after all ordering procedures have been executed. It tests each of the input options and reads data as requested. First, L is set to the input order unless option 'ILST' (use last order) has been requested. Next, the options 'IRAW,' 'IPCT,' 'ICOR,' 'IORD,' 'IORG,' 'ILST,' and 'IGRP' are processed in separate conditionals. The pattern for most of the conditionals is a page restore, reset of BASE.P and P, printing of a message to show the type of input requested, printing of the title of the data case, and listing of the order of the data. For 'IRAW' and 'IPCT' the operations are a page restore, reading the title of the new data case, setting of BASE.P as a function of option 'IADD,' reading of the raw data, computation of the proper correlation matrix, printing of a message, and printing of the title of the data case.

PERMUTATION SEARCH:

Procedures NEXT1, NEXT2 are typical procedures for generating patterns to be searched for the best seriation. They initialize themselves when the parameter SELECT is set to 0, and generate the next permutation in line when SELECT is set to 1. PSI, PSJ, and perhaps PSK are used to control the process; they are integers global to Permutation Search, but are not altered by any other procedures. As soon as each pattern is generated, the procedure interrogates option 'PSAL' to see if a report of the trial order is to be printed. NEXT1 computes permutations for the pairwise interchange pattern; NEXT2 computes permutations for the successive-rotation
pattern. When the last pattern in a sequence is generated, QUERY is set FALSE; in all other cases, QUERY is set TRUE.

Procedure PSCH.IMM controls the immediate-improvement search. In this version, it uses NEXT1 to control its search. It first initializes NEXT1 and saves the starting value of the norm in TRIAL.NORM. Next, the procedure enters a loop which runs until all patterns have been generated (i.e., until NEXT1 returns with QUERY = FALSE). In the loop, the next permutation is generated and tested, and L (the old order) is set to TRIAL. L (the generated order) if the new order is an improvement. In such cases, TRIAL.NORM is set to the new norm and a note that a new order has been accepted is printed if option 'PSTR' has been called.

Procedure PSCH.DLY controls the delayed-improvement search. It is similar to PSCH.IMM except that BEST.L is used to retain the improved ordering until the end of the cycle.

Procedure PERMUTATION SEARCH is the control procedure for Permutation Search. It selects the improvement patterns to use and executes successive cycles through both delayed and immediate-improvement search until no improvement is found in the course of a cycle or until 20 cycles have been performed. (We find that the process usually converges in three or four cycles.)

OTHER SERIATION PROGRAMS:

 Procedures MEIGHAN, DEMPSEY, ASCHER compute seriations by techniques other than Permutation Search. The programs are not included here.

INITIALIZATION AND MAIN CONTROL:

The initialization section sets up the entries in OPT.NAM and initializes the indices into the option vectors; it also initializes certain vectors to the identity permutation. The main driver loop begins at the label CYCLE. For each task, it reads control and teletype output options, sets up the line length, reads the input, calls the appropriate seriation procedures, generates output, and returns to execute another task. It also terminates the run when a 'STOP' option or the end of the input file is encountered.

Altering the System

The PHOENIX system was designed with ease of extension and alteration as a prime objective. The consequent modularity of the system makes both the addition of new features and the modification of old ones reasonably simple tasks. We shall give detailed explanations for five such modifications; this should be sufficient to indicate how to make any other desired changes.

TO ADD AN OPTION:

Assume we wish to add an option 'TIME.' The procedure is:
1) Prefix its 4-character name with 'Q' and add it to the integer declaration which contains all the other options:

QTIME, 'Q' PERFORM OPTION TIME

2) Add a line to the initialization sequence to set the integer just declared (the index into the option vectors) to some unused entry and to initialize that element of OPT.NAM to the 4-character name of the option:

QTIME: 49; OPT.NAM[49]: 'TIME';

3) At each place where the optional operation might be performed, test the appropriate element of option:

if OPTION [QTIME] then

4) If the task involves printing, send the appropriate element of TTY.OPT to procedure PRINT before the task and a zero to PRINT afterward:

if OPTION [QTIME] then

begin

PRINT (TTY.OPT[QTIME]);

body of task

PRINT(0);

end;

TO ADD A SERIATION TECHNIQUE:

Assume now that the option 'TIME' is intended to call on a new seriation procedure named DO.TIME.

1) Add the option name and index to the declaration and initialization sequences as described above.

2) Add a line to the driver between the calls on READ.INPUT and PRINT.OUTPUT to call the control procedure for the technique when the option is called:

if OPTION [QTIME] then DO.TIME;

3) Add the procedure DO.TIME and any auxiliary procedures in the section allotted to seriation procedures. They may call on any of the output support procedures, and may either declare their own variables or use the predeclared variables in accord with their original purposes. The raw data is in array RAW, the percentages are in array A, the correlation matrix is in array X, the current order is in array L, T gives the number of types, and P gives the number of sites. The procedure should return the best order found in vector L.
TO CHANGE THE NORM:

To supply a completely new norm, provide a new procedure EVAL:

```plaintext
real procedure EVAL(ARR,ORDER);
real array ARR,ORDER;
```

The procedure must give the (real) value of the norm of the correlation matrix ARR in the order given by ORDER. It should also make some provision for being called with OPTION[QTPCT] = TRUE (type-percent option of Permutation Search); in this case it should evaluate a norm directly from A.

TO CHANGE THE CORRELATION COEFFICIENT:

Assume that another option for computation of the correlation matrix (say, 'MXAA') is desired. To implement this,

1) Add 'MXAA' as an option as described above.
2) Add a procedure COMP.AA similar to COMP.BR and COMP.PM to compute the correlation matrix X from A and RAW according to the new formula AA.
3) In procedure READ_INPUT, change the command:
   ```plaintext
   if OPTION [QMXBR] then COMP.BR
   else if OPTION [QMXPM] then COMP.PM
   else COMP.BR;
   ```
   to the command
   ```plaintext
   if OPTION [QMXBR] then COMP.BR
   else if OPTION [QMXPM] then COMP.PM
   else if OPTION [QMXAA] then COMP.AA
   else COMP.BR;
   ```

TO ADD A SEARCH PATTERN:

NEXT1 and NEXT2 are examples of permutation generators. Assume that another (say NEXT3) is to be added to follow NEXT1 in the immediate-improvement portion of the search cycle. NEXT3 (OLD, NEW, SELECT) should be written to initialize itself when called with SELECT = 0 and to provide a new trial permutation when called with SELECT = 1. It should set QUERY to TRUE for all calls except the one which causes the permutation pattern to be exhausted. On that (final) call, it should set QUERY to FALSE to indicate the end of the pattern. To add the procedure,

1) Add the procedure NEXT3 in the vicinity of NEXT1 and NEXT2.
2) Duplicate the existing code for PSCH.IMM at the end of that procedure, substituting NEXT3 for NEXT1 and LOOP2 for LOOP. This will cause a cycle of NEXT3 to be searched at the end of each cycle of NEXT1.
BIBLIOGRAPHY

ADAMS, R. McC.

ASCHER, MARCIA

ASCHER, MARCIA AND ROBERT ASCHER

BELOUS, R. E.

BORDES, FRANCOIS

BRAINERD, G. W.
1951 The place of chronological ordering in archaeological analysis, American Antiquity 16: 301-313.

DEETZ, JAMES AND EDWIN DETHLEFSEN

DEMPSEY, PAUL AND MARTIN BAUMHOFF

DIXON, K. A.
1956 Archaeological objectives and artifact sorting techniques: a re-examination of the Snaketown sequence. Western Anthropology, No. 3.

DRIVER, H. E.

FLANDERS, R. E.

FORD, J. A.

FREEMAN, L. G. AND J. A. BROWN

HOLE, FRANK AND K. V. FLANNERY
Unpublished Manuscript. Phases in the paleolithic of Western Iran.

HOLE, FRANK, K. V. FLANNERY AND J. A. NEELY
1965 Early agriculture and animal husbandry in Deh Luran, Iran. Current Anthropology 6: 105-106.

In press Excavations in Deh Luran, Iran. Smithsonian Contributions to Anthropology.
Jelinek, A. J.

Kroeber, A. L.

Kuzara, R. S., G. R. Mead and K. A. Dixon

Lehmer, D. J.

Lipe, W. D.
1964 Comment on Dempsey and Baumhoff's "The statistical use of artifact distributions to establish chronological sequence." American Antiquity 30:103-104.

Meighan, C. W.

Petrie, W. M. F.

Proskouriakoff, Tatiana

Robinson, W. S.

Robinson, W. S. and G. W. Brainerd

Rowe, J. H.

Troike, R. C.

Willey, G. R.