DOCUMENT RESUME

ED 062 653	24	CG 007 176

AUTHOR	Kadushin, Charles
TITLE	Sociometric Clique Identification. Final Report.
INSTITUTION	Columbia Univ., New York, N.Y.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C.
BUREAU NO	BR-9-B-022
PUB DATE	Jul 70
GRANT	OEG-6-2-9-40022-1020
NOTE	65p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Behavioral Sciences; Computer Graphics; *Computer Oriented Programs; *Data Processing; Electronic Data Processing; *Information Processing; Social Behavior; Social Factors; Sociocultural Patterns; *Sociology; *Sociometric Techniques

ABSTRACT

This report consists of four parts. The first part is a non-technical summary of the basic problem and an attempted solution. The second part is a technical review of the literature and a description of the basic algorithm used in the solution. The third part describes the use of the Sociogram System. The fourth part describes the use of CHAIN, a program for discovery of sociometric linkages. There is a discussion of a solution to the sociometric clique identification problem by the use of an efficient method of generating sociograms and an efficient computer-based method is presented. This method is composed of two steps. In the first, a matrix of pairwise relatedness is calculated. In the second step, a multi-dimensional scaling technique is used to generate the configuration is then displayed on a cathode ray tube plotter and the lenis are drawn where a relationship exists. (Author/BW)

4-13-72

Approved 5/31/72

John Sokol

Mr. John Sokol
Director, Educational Research
DHEW-OFFICE OF EDUCATION - RM 1013
Federal Building
26 Federal Plaza
New York, New York 10007

OE-NCE

FINAL REPORT
Project No. 9-B-022
Grant No. OE6-2-9-40022-1020

SOCIOMETRIC CLIQUE IDENTIFICATION

Charles Kadushin
Teachers College, Columbia University
525 West 120 Street
New York, New York, 10027

July 1970

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

Final Report

Project No. 28-5476
Grant No. OE6-2-9-40022-1020

Sociometric Clique Identification

Charles Kadushin

Teachers College, Columbia University

New York, New York

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Research

PREFACE

This report consists of four parts. The first part is a non-technical summary of the basic problem and our attempted solution, written by Charles Kadushin. The second part is a technical review of the literature and a description of the basic algorithm used in our solution, and was written by Richard Alba and Charles Kadushin. The third part describes the use of the Sociogram System and was written by Richard Alba, chief programmer for this project, who has also been responsible for developing our basic algorithm. The fourth part describes the use of CHAIN, a program for discovering sociometric linkages, and was written by Peter Abrams and Richard Rosen. Although the various sections of the report have been the particular responsibility of the persons given credit for them, all of us have participated in extensive discussion on all phases of the work.

CONTENTS

SOCIOMETRIC CLIQUE IDENTIFICATION

INTRODUCTION

Sociometry

Sociometry is a method for ascertaining the relationship between units. Usually, the presence or absence of a relationship of one unit to all other units is mapped. Sometimes the relationship mapped is one of degree. Typical relations that have been studied are the liking or disliking of people for each other, the admiration in which they hold each other, and with whom they talk, communicate, or otherwise interact. Although the units studied have most often been people, especially school children, the method is obviously not limited to individuals. Relationships between different organizations or parts of an organization, as well as between cities and countries, have also been studied.

Once the relationship between all the units under study has been ascertained, a graph called a sociogram displays the network of relations. The familiar road map is one kind of sociogram, for it displays the relationship between cities or other points on a map. The line which connects the cities is proportionate in length to the distance between them. A map or sociogram not only gives an instant visual image of the network between units, but also can suggest clusterings of the units into cliques or regions. Further, the network displays an individual unit not only as having direct relations with other units, but also as having indirect relations *via* other units.

The collection of systematic data on networks of relations between units allows for many mathematical operations that can enhance our understanding of the network (Abelson, 1966). The simplest operations allow for the location of "stars" or "sinks". These are units into which many relationships flow. The most popular child in a class is a star. Or one may compute the reverse, that is, units which have many relationships which lead away from them. An effective social climber is one kind of such unit, for he seeks out many persons. Some units may have many direct relations with others, while some units may have many indirect relations. For example, a person may have a great many friends, none of whom has much consequence or power. Another person may have one good friend, but that friend may know many other well connected people who, in turn, are well connected. Thus, the ratio of direct to indirect links for each individual may be computed. These are but some of the many operations that can be performed on networks.

Since sociology, social psychology and the study of educational organizations always imply the study of social relations, it would seem that sociometry would be a widely used tool. Because of the ease with which data can be collected in school systems, education studies are probably the largest single source of sociometric data. In fact, sociometry was invented to deal with the problems of a corrective institution for adolescent girls.

Despite the apparent power of sociometry to deal with problems of group relations, the method has been a tool ahead of its time. The vast amount of sociometric data which has been collected has rarely been properly analyzed. It is impossible to hand draw sociograms of groups much larger than 25 individual units. In fact, most larger sized sociograms that we have inspected and compared, when possible, with the original data, contain at least one error. Not only is it impossible to draw sociograms by hand, but it is equally impossible to assign units to cliques or regions. The juggling involved seems too difficult for an unaided human. Because of these very practical difficulties, sociometry has never been able to fulfill its promise.

There is a long history of attempts to solve the problem of clustering. Beginning in 1949 (Luce and Perry), various matrix algebra techniques were introduced in attempts to simplify the clique formation problem through mathematical reduction of the data. As late as 1965 (Harary, Norman and Cartwright), attempts were made to apply graph theory to sociometry, since the sociogram itself is one kind of graph. For technical reasons which Spilerman (1966) and Abelson (1966) give, and which are elucidated in Alba and Kadushin (1970) which forms part of this report, sheer mathematical methods are simply not able to cope with the complexity of sociometric data. Some iterative algorithm (a rule of procedure which is repeated over and over again with each approximation to the answer somewhat better than the previous one) might be more helpful.

Even before high speed digital computers with fairly large memories were developed, various investigators proposed several algorithms with which card sorting accounting machines could rearrange sociometric data so that cliques might be more readily seen. A recent approach is Spilerman's (1966) method, which uses a large, high speed digital computer. But even this modern attempt is disappointing. It is effective only if there are relatively few connections between units. Moreover, the computer program itself has several "bugs". The most serious problem, of which the author of the program was unaware, is the fact that different data arrangements would be produced depending on where in the data the program started to operate.

Toward a Solution of the Sociometric Problem

It is safe to say that at the time we began our work the sociometric problem still defied solution. Nonetheless, the basic tools for solving the problem seemed, for the first time in 30 years, to be lying about merely waiting for someone to put them together. To begin with, Columbia University had just acquired one of the world's largest computers (an IBM Model 360-91 coupled to a 360-75), as well as a data plotter. More important, recent work in non-matric scaling and in numerical taxonomy (see Green, et al. 1968 for a relatively non-technical review) suggested

that many of the clustering and display problems that have proved so irksome to social science were in the process of being solved. Today, in 1970, the entire field of non-metric clustering is less than ten years old. We thought that clustering programs could be enlarged with the aid of our new mammoth computer and adapted to sociometric analysis.

There were several hurdles in the way. The most crucial problem was the fact that most sociometric data was, and still is, collected as dichotomous data; that is, a unit either is or is not related to another unit. No known clustering technique can work with such data - in fact, this was the main problem in previous attempts to cluster sociometric data, even before the advent of new non-metric techniques. Our basic insight into this barrier came from the theory of social circles (Kadushin, 1968). Neither in actual life nor in artificial sociograms is a unit merely related to other units directly. The essence of sociometric analysis is that it shows that units may be indirectly related to other units. A is related to B who, in turn, is related to C. We had previously argued that this simple fact is a key to the way modern social systems operate. But this fact could also be used to create a measure of relatedness between pairs of units so that these units could then be grouped with some of the new approaches to clustering. A is one step away from B and two steps away from C. In fact, several previous mathematical analyses of sociograms had pointed this out but were not able to connect it to clustering techniques, because these clustering techniques had not yet been invented.

In order to make the best possible use of the idea of the number of steps from one unit to another, we investigated a computer program developed by James Coleman (1964) which was designed to show how many persons in a group were "ultimately", that is, even through a very long indirect chain, connected to all other members of the group. Thus, instead of showing no connection between A and C, the program would show a connection if they were both directly connected to B. The measure of "connectedness" he developed was simply the number of actual connections divided by the number of possible connections for any given group. The program was also able to show the shortest number of steps from one unit to another. Further, the program gave a "printout" of all units which were even remotely connected to a starting unit. This printout is a very useful research tool in "debugging" sociometric data. The only problem with the program was that it did not work, and contained errors in calculations even when the programming errors were straightened out!

Our first job was to correct Coleman's program and make it generally available for third generation computers (it had been written for a second generation computer). CHAIN (see page 46 following) describes this program. Those familiar with Coleman's program, as printed in his book (1964), will note a number of new features which add to the program's flexibility, both in data handling and data analysis. Together with RENUM (pages 27-31),

almost any kind of input data can be handled and the basic part of the program allows for the generation of data that can be used for a large variety of sociometric data analysis purposes.

Our original purpose in developing CHAIN was to produce a matrix which gave the shortest path from one unit to another. This matrix was to be clustered with the aid of Johnson's hierarchical clustering program (pages 43-44). This program clusters all units which are not more than a given distance away from one another (Diameter method). Thus, all units which are one step away from each other are clustered, then all units which are one and two units away from each other are clustered, then all units which are one, two and three units away from each other are clustered, and so on. In addition, in its connectedness method, the program groups units which can be reached from a given starting unit. Thus, if we start with unit A, a unit which is one step away from A is added to the cluster, then a unit which is one step away from the unit just added is included, and so on. Both these methods seemed ideal for clustering sociometric "distance matrixes" as produced by CHAIN.

The only problem was that no clustering method could work with the data which CHAIN produced. The matrix of shortest distance from one unit to another led to many ties. There were many units which were only one step away from other units, many which were two steps, and so on. Almost all clustering programs are confused by a proliferation of ties. Some algorithm for relatedness would have to be found which did not lead to many ties. Further, it was also discovered through the process of laboriously checking out the results of clustering the shortest distance matrix, that some method of accurately drawing sociograms by computer would have to be developed <u>before</u> any further work with clustering techniques. Existing hand drawn sociograms proved to contain errors, so that they could not be used to verify computer produced clusters. Without an accurate, easy to read sociogram, large numbers of different types of sociometric clusters could not be compared with the original data, and for the development of effective clustering techniques, it is necessary to work with many different types of sociograms. The drawback of previous efforts was that they worked with the data on hand, but proved unable to cope with other sociometric data.

Because of these problems we started all over. First we constructed a new algorithm (described in detail in the appendix to Alba and Kadushin, pages 19-22 below) for measuring relatedness. This algorithm uses much more of the data, for it considers not only the shortest paths, but all non-redundant paths between any two units. Longer paths count for less, however, than shorter paths. Because so much more of that data is used, the chances of ties are quite low. In addition, the algorithm is likely to locate units in space relative to each other in a way which is much more like the original data.

Now that we had a matrix of relatedness which gave an almost infinite variety of distances of one unit from another, we used a non-metric scaling technique which located units in two or more dimensional space such that

the rank order of the distance between units in the space created by the program was roughly the same as the rank order of the distances between units in the original data. The problem is something like being given a mileage chart between cities of an unknown country and being asked to construct a map based on the mileage chart. It sounds easy, but once more, this is the sort of problem which humans are essentially incapable of solving on their own.

Kruskal's non-metric scaling program (pages 36-38 below) located our units in two dimensional space. It remained only to draw lines between them which, as in conventional sociograms, represent direct choices. To do this we developed a plotting program which plots the points, labels them, and draws the appropriate lines representing the direct choices. For the first time since sociometry was invented, it was possible to obtain accurate and objective sociograms. Sample sociograms are shown in Alba and Kadushin, pages 10-15. They come from a study of opinion makers. Although the purpose of the study was to develop programs for the study of schools, opinion maker data were on hand and seemed unusually complex. If we could produce satisfactory sociograms for this data, surely we could do so for school systems. The program has, in fact, recently been used for school studies and has proved most satisfactory.

It is rare in social science that there is an objective goal - either one is right or one is wrong. In our case, we had a definite goal - to produce computer drawn, objective sociograms - something no one had ever been able to do before. We succeeded. Nonetheless, this report is a half-way mark. The ultimate goals of, first, objective clique identification and, second, the production of fully documented computer programs adaptable to different installations, have not yet been attained. And yet we feel that we have solved the problems in principle: that our algorithm for finding relatedness between pairs effectively produces a measure that can be scaled to produce sociograms, or clustered to produce cliques. We feel that we have generally identified the class of scaling and non-metric clustering techniques that will work with this algorithm. Our set of computer programs to produce sociograms and perform rudimentary clique analysis is wholly operational at Columbia University and can be used by persons with no knowledge of programming. The Columbia University Computer Center, which participated in the development of the program, will give access to any academic groups interested in using our program.

In keeping with the nature of our progress, then, this report consists of three parts. The first is a paper submitted to Sociometry. The second consists of detailed instructions for using the set of programs at the Columbia Computer Center. All the programs except the relatedness program can be used at most large computer installations. The relatedness program will operate only with very large IBM computers. The third part of the report consists of a program write-up of CHAIN, a program which finds the shortest number of steps between two points of a sociogram. This program was useful in our early work, and represents a correct version of a program published by Coleman.

Despite the incomplete nature of our work, it is my feeling as principal investigator, that we have written more programs of a more sophisticated nature and solved more problems than is usual for projects of this size.

Abstract

The solution of the sociometric clique identification problem could be greatly advanced if an efficient method of generating sociograms could be invented. An efficient, computer-based method is presented in this paper. This method is composed of two steps. In the first, a matrix of pairwise relatedness is calculated; the measure of relatedness depends upon the number and lengths of paths from one point to another in a directed graph. In the second step, a multi-dimensional scaling technique is used to generate the configuration of points in space whose interpoint distances best monotonically match the measures of pairwise relatedness. This configuration is then displayed on a cathode ray tube plotter and the lines are drawn where a relationship exists by virtue of direct nomination; the result is interpreted as a sociogram.

THE CONSTRUCTION OF SOCIOGRAMS BY COMPUTER METHODS[1]

Richard D. Alba and Charles Kadushin
Columbia University and Teachers College, Columbia University

Although sociometry was once associated with a particular psychological theory (Moreno, 1934), it has come to mean any method for collecting and analyzing patters of relations between individual units. Typically, a sociometric question on a survey asks a respondent to name those people who bear a particular relation to him: those he likes, those whom he asks advice, and so on. An observation as coded for sociometric analysis usually consists of an identification of the respondent together with an identification of the persons to whom he is related.

Sociometry has had wide application outside the study of small groups where it started. It has been used to study interlocking boards of directors (Guttsman, 1963), informal networks among members of an elite (Kadushin, 1968; Agger, 1964), relations between community organizations (Young and Larson, 1965), "invisible colleges" among scientists (Mullin, 1968; Crane, 1969), and the diffusion of new practices among physicians (Menzel, Coleman and Katz, 1966). Perhaps the most frequent application has been to studies of clique formation among school children (Coleman, 1961).

There is a large body of literature, some of which will be presented in the following section, which addresses the problem of identifying cliques and subgroups among individuals in a large group. A clique may be "intuitively defined as a subset of members who are more closely identified with one another than they are with the remaining members of their group" (Hubbel, 1965: 377). The identification problem is unsolved.

In very general terms, there are two ways by which to handle the problem of clique identification. The first is to devise some numerical or mathematical rule by which likely subsets of individuals can be identified; ideally this rule can be incorporated in a computer program and large bodies of data can be efficiently handled. The second is to present the data in a form, such as the sociogram, which is isomorphic to the original data and yet which increases the visibility of cohesive subgroupings. The researcher may then identify cliques in a manner which is intuitively satisfying.

---

[1]This research was supported by a grant to Charles Kadushin by the grants program of the Office of Education. The authors wish to thank Kenneth King, Director of the Columbia Computer Center, and Dr. Charles Roberts of Bell Laboratories for their help and encouragement. Richard Rosen materially advanced the research with some early conceptualization and experimentation. The help of Peter Abrams has been invaluable.

The difficulty with the first method, that of identifying subsets of individuals by a numeral or mathematical rule, is that the resultant cliques are not necessarily the most intuitively satisfying ones. The difficulty with the second method is that sociograms must be constructed by hand and, for groups of any size, such constructions are tedious at best.

If an efficient means of generating sociograms could be invented, then the problems of validating mathematically-based methods of clique identification would be considerably eased. Spilerman's comments are germane:

> One way by which the researcher could investigate the biases inherent in a prespecifying routine would be to compare its decisions with his perceptions regarding the structural components present in the data. For example, he might construct the sociogram for a portion of the group and select which groupings he wants identified as cliques, which definitely are not cliques, and which groupings appear ambiguous. Then, any objective procedure which agrees with his first two decisions could be allowed to classify the ambiguous groupings and to process the remaining data with a reasonable assurance that it will not grossly transgress his perceptions.
>
> For this reason it seems likely that the provision of efficient means for representing group structure will facilitate the empirical validation and adoption of more sophisticated mathematical techniques. (Spilerman, 1966: 313)

This paper will present an efficient, computer-based method of generating sociograms. The authors believe that they have produced the first computer-drawn sociograms. The reader should be warned that the programs which have produced these sociograms are not yet ready for general distribution. However, the algorithm and method are presented in detail in this paper.

Previous Work on Clique Identification

In the previous section, we mentioned two possible approaches to clique identification: the first is that of mathematical techniques; the second is that of isomorphic presentation, such as the sociogram.

A sociogram is a pictorial representation composed of points and lines connecting points where each point represents an individual and a line, which may have an arrow to indicate direction, connects two points if the individuals represented by those points are related by virtue of one having chosen the other or each having chosen the other.

While a sociogram is one kind of representation of sociometric data, there are two other kinds which are important because they connect sociometry with powerful sets of mathematical theory. The first of these is the directed graph; the theory of directed graphs is a formal way of talking about some important features of networks. A directed graph, itself, is best visualized as a directed picture such as a sociogram. It is defined formally as a set of points together with a set of ordered pairs, whose elements belong to the set of points. The second representation of sociometric data is the adjacency matrix A. The elements of the adjacency matrix indicate direct relationship or lack of it; the entry a_{ij}, the entry in the ith row and the jth column of A, is 1 if individual i names, or is directly related to, individual j and is 0 otherwise. We can therefore bring to bear on the problems of sociometric analysis two powerful sets of mathematical theories: those of matrix algebra and those of directed graphs.

Nearly all mathematical techniques of clique identification have drawn from one or both of these sets of theory. One such group of techniques is composed of those in which a formal definition of a clique is presented; such a formal definition is usually stated in terms of the properties of adjacency matrices, matrices derivable from adjacency matrices, or directed graphs.

Foremost among the methods which use a formal definition of cliques are those presented in Luce and Perry (1949) and in Harary, Norman, and Cartwright (1965). Both methods appeal to concepts in directed graph theory to define a clique and then use operations on adjacency matrices to locate subsets of the entire group which satisfy the definition of a clique. (It should be noted that Harary, et.al., never actually use the term "clique" but their treatment of certain topics in directed graph theory, such as strong components, implies a definition.)

A concept from directed graph theory which is fundamental to any approach to clique identification is that of reachability. Intuitively, one individual is reachable from a second if there exists some chain of individuals by which the second can communicate to the first; for example, if a talks to b, and b talks to c, then c is reachable from a. While reachability is a property of points in a directed graph or individuals in a network, the reachability relations of a directed graph can be determined from the various powers of the adjacency matrix isomorphic to it. Non-zero elements in the square of the adjacency matrix, to take an example, occur only for individuals or points which are connected by a chain of two links.

Luce and Perry (1949) propose using the third power of the adjacency matrix to identify subsets of individuals which might be cliques. As Spilerman (1966) points out, this method may fail to produce common sense cliques. According to Abelson (1966), there is no reason to think that chains of any given length (say, 3) are more indicative of cliques than chains of any other length.

The concept of strong component, as presented in Harary, Norman, and Cartwright (1965), seems appealing as a possible definition of clique. Two individuals are in the same strong component if each is reachable from the other. When we condense a directed graph by replacing each of its strong components by a single point, the resulting condensed directed graph preserves the essential reachability features of the original. The problem seems the same: if we take strong components to be cliques, the resultant cliques may not have great intuitive appeal. Moreover, when the relation is symmetric, such as "talk with," entire networks may disappear into a single point when condensed.

Other writers have presented ways of manipulating the data so that the cliques become more visible to the eye, even though the data is presented in a form isomorphic to the original, that is, with all the detail of the original. Coleman and MacRae (1960) present a method for reordering the adjacency matrix in such a way that cliques and cohesive subgroups appear as clusters or clumps along the diagonal. As Spilerman (1966) points out, these clusterings appear only under ideal conditions. For reasonably complex networks, the clique clusters along the diagonal have "octopus arms" extending out from them which confound the interpretation of results.

Spilerman (1966) attempts to combine the best features of the Coleman and MacRae method with those of sociogrammatic presentation. He presents a method to facilitate the construction of a sociogram from the adjacency matrix. His method generates a series of linear chains from the adjacency matrix; these linear chains are a decomposition of the sociogram which may be constructed by hand from them. While his method would seem to work well with structures which are basically chains (such as the example from the Adolescent Society in Figure 2), it is not clear that it would yield satisfactory results with structures which are more complex (such as either of the examples from the Yugoslav data).

Hubbel's method (1965) represents an important strand in the body of mathematical techniques of clique identification. His method is an attempt to cluster points on the basis of a matrix whose elements measure the relatedness of individuals in a network.

The first step in his method is to prepare the matrix of relatedness. He starts from a matrix W whose elements measure the degree of immediate relatedness between individuals. To measure the degree of indirect relatedness, that is, relatedness across chains of some given length, say p, the matrix W need only be raised to the pth power. The matrix Y which is the sum of W and its power matrices is the matrix of relatedness which we will use to cluster the points; each element of Y measures the degree of total relatedness (total in that it is based on all direct and indirect relations) for some pair of points.

The second step is to cluster the points based upon the values of the elements in Y so that clusters of highly interrelated points emerge. Hubbel suggests collecting those dyads (or pairs) for which the corresponding element of Y is above some arbitrary threshhold value. Cliques are then constructed from these dyads

Of course, the same objection which has been raised against other mathematical techniques may be raised against this one: namely, the resulting cliques are not necessarily the ones the researcher would have identified had he been able to visualize the network as a sociogram. There is another important question which may be raised: Are there more appropriate clustering methods than the one used by Hubbel?

A Proposal for the Computer Construction of Sociograms

The method which we are about to describe comes closer to the method of Hubbel that to any of the other methods we discussed in the review of the literature; that is, it is composed of two steps: the first is one in which a matrix of relatedness is derived; the second is one in which the points are clustered based upon the matrix or relatedness. It is very important to observe that these two steps are the crux of any modern solution to sociometric clique identification. It can be shown that data with a large number of ties, especially dichotomous data such as choosing a person or not choosing a person, cannot be clustered except under special circumstances. A measure of relatedness between persons which can assume a relatively large number of different values must therefore be developed. Such a measure might be derived from respondent's ratings of other persons (Hubbel, 1965; Abelson, 1966). Because the collection of such data can be quite cumbersome in large groups, our approach is to derive the measure from simple dichotomous choice data. Once a satisfactory measure of relatedness is derived, then a variety of new and powerful numerical clustering methods become applicable. Further, the most immediately visible way in which our method differs from other attempts to cluster sociometric data is that, rather than rest with emergent clusters of points as cliques, lines are drawn, after clustering, between points where a relationship exists by virtue of nomination and the result is interpreted as a sociogram. The authors do not contend that their particular method is a priori the best one, but only that it has yielded significant results on bodies of data to which it has been applied and thus may point a way to a general class of sociometric clique identification methods.

It seems reasonable to base the measure of relatedness on the graph properties of the network; that is, the relatedness of any pair of points is measured by the number of chains between the points and the lengths, or numbers of links, in these chains. Thus, we could count the chains or paths between the points, weight each chain according to length so that the longer chains count less, and then sum the weights to arrive at our measure.

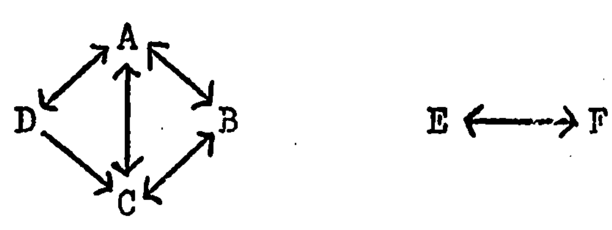


Figure 1

Thus, in Figure 1, there is a one 1-link chain from A to B (A B), one 2-link chain from A to B (A C B), and several 3-link chains from A to B (one such is A D C B). In one of the 3-link chains from A to B, namely, A B A B, a link appears more than once in the chain; such a chain, in which the same link appears more than once, is a redundant chain. On the other hand, there is one 1-link chain from E to F; and there is one redundant 3-link chain (E F E F). In terms of our relatedness measure, then, A is more closely related to B than is E to F.

We have already noted that there is a relation between the number of chains between any two points in a directed graph and tha values of corresponding elements in the powers of the adjacency matrix; that is, we can determine from A^p the number of chains of length p between any two points. Moreover, we can express the idea of attenuation, that is, that chains of increasing length count for less, by scalar multiplication of A^p by a^p, where a is a positive number less than 1.

Utilizing these two ideas, we can describe the matrix of relatedness, R, as follows:

$$R = aA + a^2A^2 + a^3A^3 + \ldots + a^nA^n + \ldots$$

Having described the relatedness measure in terms of matrix operations on the adjacency matrix, our measure is easily programmed on a computer.

One minor difficulty is that in counting chains by iteratively raising the adjacency matrix to all possible powers, we also count redundant chains. Redundant chains can be eliminated by utilizing a suggestion of Coleman (Coleman, 1964), in which elements of the adjacency matrix are set to zero when the links which they represent are used to construct a chain, thereby preventing the reuse of these links.* Moreover, this zeroing out procedure makes the process of finding all possible chains a finite one. The exact algorithm we use is described we use is described in the appendix.

There are two basic ways by which a relatedness matrix may be made comprehensible. In one, the points are placed into N dimensional space; in the other, the points are ordered into a taxonomy. The first corresponds to presenting the matrix as a sociogram; the second to placing individuals into cliques. Until very recently, factor analysis and discriminant analysis were the only practical techniques for approximating either goal. Recently developed numerical techniques (Green, Carmone, and Robinson, 1968; Ball, 1965) seem better suited, however, to the unknown properties of a relatedness measure. The

---

*Another idea implied by Coleman as a way of avoiding redundancy is to use the length of the shortest chain as the measure of relatedness. This method requires no weighting and is intuitively pleasing. Experimentation with it suggested, however, that not enough information is used to form adequate graphs or cliques if the data are complex. See Richard Rosen and Peter Abrams, CHAIN, 1970.

clustering method which we use is non-metric multidimensional scaling* (Kruskal, 1964), which, as its name implies, does not assume that our measure satisfies metric properties.

Furthermore, a computer-based method for generating sociograms has higher priority than a computer-based method for identifying cliques, since the adequacy of the latter can be evaluated by visual comparison with the former. Accordingly, the method proposed differs from other attempts to cluster sociometric data in that after the points have been placed in two-dimensional space, lines are drawn between points where a relationship exists by virtue of direct nomination. The result is interpreted as a sociogram.

Of course, a procedure of placing points randomly in space and drawing lines where a direct relationship exists would only infrequently result in a comprehensible sociogram. For an arbitrary configuration of points to result in a satisfactory sociogram, it is necessary (although probably not sufficient) that points which are directly related be closer in space than points which are not. More generally, it is necessary that the interpoint distances reflect the interpoint relatedness; that is, the more related two points are, the closer they should be in space.

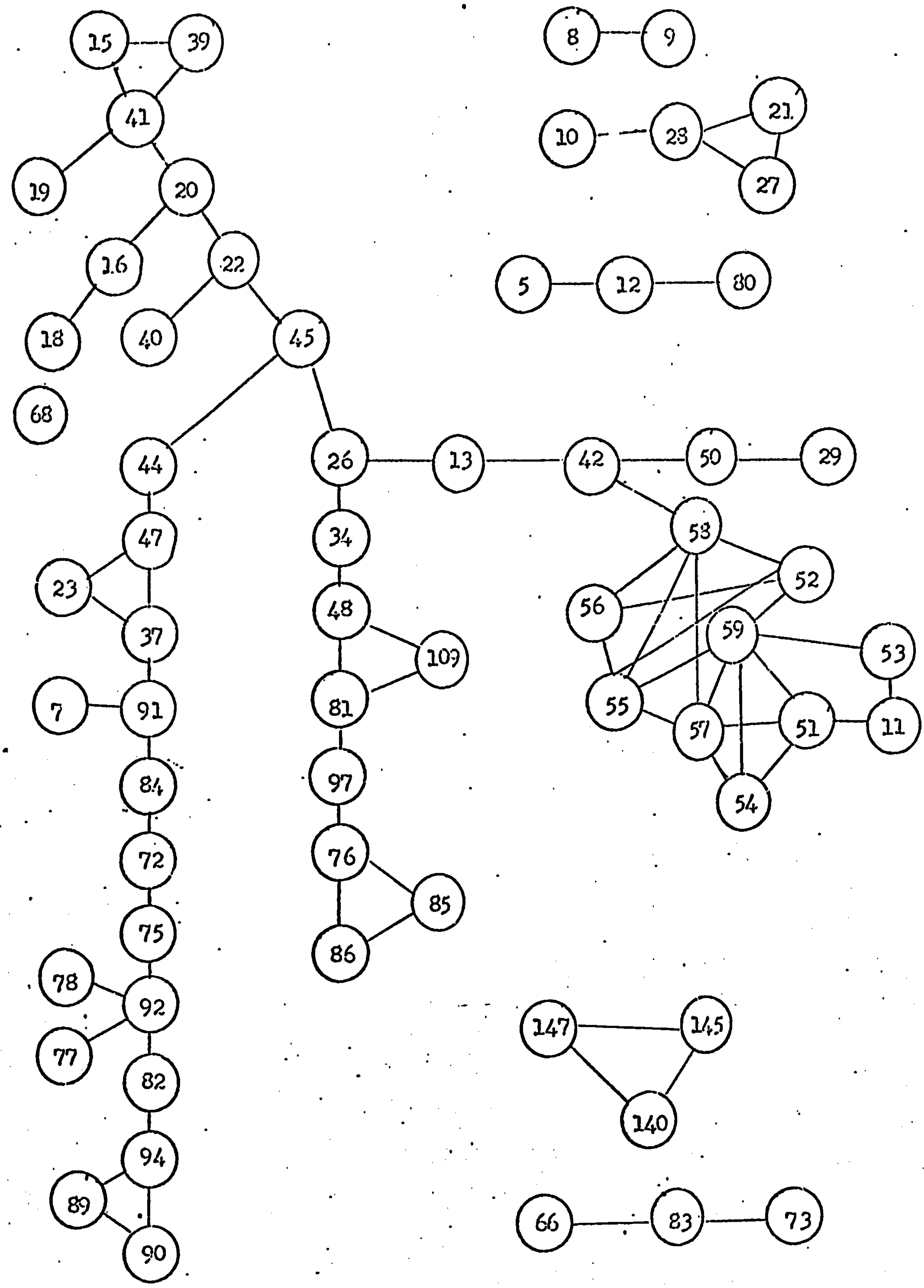
This last requirement is the rationale for our selection of multidimensional scaling as a clustering technique. Multidimensional scaling is a numerical technique which, given a matrix of similarity (or dissimilarity) measurements, constructs the configuration of points in space whose matrix of interpoint distances comes closest to monotonically matching the matrix of similarities. If we take the matrix of similarities to be our matrix of relatedness, then the more similar or related two points are, the closer we would expect them to appear in space after scaling.

To generate sociograms the points are first separated into disjoint sets, that is, sets such that no point in a set is related to any point in another set. Out programs then calculate a relatedness matrix for each set based upon a symmetric adjacency matrix. The points of each set are scaled into two-dimensional space and the resultant configuration is displayed on a cathode ray tube plotter and lines are drawn by the plotting program between points where appropriate. Some of the results are shown below. Figure 4 is a complete reproduction of a hand drawn previously published sociogram (Figure 2) which is a "chain". Only those points which are connected to each other in the large chain are shown, since the remaining five sets of units are trivial. The chart in Figure 3 lists the correspondence between the numbers in Coleman's original chart and the consecutive renumbering performed by our program.

*We wish to thank J.B.Kruskal, S.C. Johnson, and M. Wish of Bell Laboratories for making MDSCAL IV available to us and for consulting with us on various problems on non-metric techniques.

FIGURE 1

SEGMENT FROM FIGURE 7.2.1 IN ADOLESCENT SOCIETY



Dictionary of Correspondence Between Figures 2 and 4

Identification in Figure 4	Identification in Figure 2	Identification in Figure 4	Identification in Figure 2
1	15	25	90
2	39	26	26
3	41	27	34
4	20	28	48
5	16	29	109
6	18	30	81
7	22	31	97
8	45	32	76
9	44	33	85
10	47	34	86
11	23	35	13
12	37	36	42
13	91	37	50
14	7	38	29
15	84	39	58
16	72	40	56
17	72 (error)	41	55
18	75	42	57
19	92	43	59
20	78	44	52
21	77	45	53
22	82	46	11
23	94	47	51
24	89	48	54

All other points in figure 2 are omitted from figure 4

Figure 3

The points on the computer drawn sociogram are indicated by a small dot on the lines with the identification number adjacent. In this scale, the points are better seen by noting inflections of the lines. In this scale, the tight clusters of 58; 56; 55; 59; etc. (in the original numbers) in compressed beyond legibility in an 8 x 11 format, a problem that can in part be solved by "blowing up" the diagram (the program itself can do this) and in part by a method to be discussed shortly. It should also be noted that we made an error in transcribing the original data to punch cards. This is shown by the extra point 17 in the computer drawn diagram. We have preserved this error because it demonstrates our original reason for producing computer drawn sociograms: in every case of large sociograms in which we possessed the original punch card data and the hand drawn sociogram of the original investigator, we discovered at least one discrepancy between the punch card and the original sociogram.

Figure 5 represents the reported interaction connections between members of the two major Federal Yugoslav Legislative bodies in 1968, and is introduced to show the utility of our approach for large scale open system sociometric analyses. Again, the sociogram only shows the largest set of ultimately connected members. Analysis of the sociogram will be reported elsewhere (Barton, Denitch and Kadushin, 1970). It may be noted here, however, that the central cluster around members 9, 21, 48, etc. includes the chief formal officers of the legislative bodies, those whom experts had independently rated as key figures, as well as a key figure in the Yugoslav League (Party) of Communists. The quadrants are actually formed by members of different Yugoslav nationalities.

By now it should be apparent that certain features of the sociograms produced by our method differ from those which have been hand drawn. In traditional sociograms the length of lines connecting points is entirely arbitrary and generally dependent only on the requirements of demonstrating the formation of cliques. In our algorithm, the relatedness between two points depends on the number of other points they are connected with, the number of paths to these points, and the number of points in the chain between points. To the extent that the MDSCAL algorithm is able to preserve the relative distance between points as represented by the way we measure the relatedness of points, to that extent the length of lines has some meaning. So in general, the short lines in the middle of the sociogram in Figure 5 suggest that the average degree of relatedness between the points in the middle is higher than the relatedness between points at the periphery. The large loops, say 44,33,24,28,39,11,8,51,5, 41, and back to 44, are caused by the fact that to get from any one point on the loop to any other point one generally has to go through a large number of other points, and hence the relatedness is generally low and the "distance" from any point to any other is relatively great, so the size of the loop on the graph is quite large. This feature also tends to force so-called sociometric stars toward the center of the graph. Thus, if there appear to be several cliques in a graph their starts will not be at the center of each clique, but will appear more toward the center of the entire graph. This feature is caused by the tendency of

the multidimensional scaling algorithm to locate the zero point of the coordinate system at the center of gravity of the configuration. Similarly, points which have few connections to anyone "important" are "exploded" to the periphery of the graph.

Relative to the points on the periphery, the points which are highly related may be overly compressed. To see how this compression may happen, let us suppose that we have noted a configuration for all points but one (the scaling program does not actually work this way). The relatedness measures of this point to all other points will serve as a series of constraints forcing this point to a unique location. We can visualize these constraints as an array of forces acting in many directions on the point; if the relatedness matrix is completely consistent with the two-dimensional configuration, then the point will be moved by these forces until a spot is reached in which they are all zero; in general, however, this consistency will not hold and the point will be moved until a minimum point is reached where the sum of the forces is a minimum and the forces in one direction are offset by the forces in the opposite direction, as well.

To visualize how a sociogram produced by this method may be distorted by inconsistency between the relatedness matrix and two-dimensional space, let us examine Figure 6, which portrays the connected points of Yugoslav mass organization leaders. As might be expected from the history of Yugoslavia, this group is unusually dense. Hence in this figure, the part of the sociogram marked with a dotted line is compressed beyond readability.

The genesis of the problem can perhaps most easily be visualized by imagining the points in the compressed area as wanting to push the long arms out to the right, while the relatedness of these arms to the structure on the left tends to pull the arms to the left. In this push and pull, the weight of the arms is sufficient to overwhelm and compress part of the structure. In this case, a solution to the problem is to set the relatedness measures of the arms to the remaining structure to zero, allowing the arms to "float free" (of course, a few of these relatedness measures are preserved so that the general orientations of the arms remain). The resulting sociogram is shown in Figure 7.

Conclusion

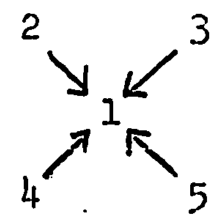
While a number of problems remain to be solved, it appears that new numerical methods along with a re-evaluation of the meaning of sociometric relatedness now promise a solution to the problem of finding intuitively satisfying cliques in large sociograms. Indeed, the size of the sociograms is limited only by the ingenuity of programmers and the size of the research budget (at present, big matrices are costly to handle). It seems to us that the following steps in the development of the study of informal structures of society are now all but inevitable: (1) solution of the "compression" problem; (2) the graphing of large numbers of sociograms in a wide body of data; (3) the generation of more

efficient techniques for such graphing, including the exploration of the effect of various algorithms for finding relatedness; (4) the generation of adequate numerical hierarchical clustering methods for sociograms and their testing on wide bodies of data; (5) the development of better global descriptions of large sociograms, that is, the development of typologies of sociograms; and most important (6) the development of theories of informal structures which relate typologies of structure to other relevant social facts such as typologies of formal structures.

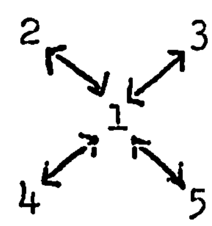
Appendix. The details of the algorithm

Although the distances between points in a sociogram have not been defined as meaningful, we start from the recognition that any pictorial representation involves constructing a configuration of points in a metric space, and hence distances emerge as a byproduct. This recognition led to the generating idea of our technique: the use of non-metric multi-dimensional scaling to generate a configuration of points in the plane.

Having decided to use multi-dimensional scaling to construct the configuration we will interpret as a sociogram, we must define some appropriate similarity or dissimilarity measure. It seems reasonable to base this measure on the number of nonredundant chains* between points and the lengths, or numbers of links, in these chains.



(A)



(B)

Figure 8

Several minor difficulties should be noted. Figure 8 shows two sociograms which require the same configuration of points in space. In sociogram A, all relationships are unreciprocated or asymmetric; moreover, there are no chains between any two of the points: 2, 3, 4, 5. Therefore, our similarity measure is zero for any pair of points drawn from this set; in other words, we have no information as to how these points should be arranged in space. In sociogram B, however, all relationships are reciprocated. Examination shows that our measure is non-zero for any pair of points in the sociogram. Comparison of

*We use the term "chain" where Harary, Norman, and Cartwright (1965) use the term "sequence". Our notion of nonredundant chain corresponds loosely to their notion of path, although our nonredundant chain is more inclusive than their path, that is, not every nonredundant chain is a path, while every path is a nonredundant chain. Luce and Perry (1949) use the term "chain" in a way which corresponds to our use. In their paper, the term "nonredundant chain" corresponds exactly to "path", however.

these two sociograms suggests that we obtain more information for the purpose of locating points in space when we reciprocate all direct relations. In the following discussion we will assume that we are working with a "symmetricized" relation.

In addition to symmetricizing the data, we partition it into disjoint subsets such that: no point in any set is reachable from any point in any other set; and any point in any set is reachable from every other point in that set. If we did not partition the set in this manner, it is quite likely that there would be at least two points which were not reachable from each other; then, the similarity measures would be zero for these points and we would have no information about the distances between the points in space.

We now present certain known results for directed graphs (Coleman, 1964; Harary, Norman and Cartwright, 1965). If A is the adjacency matrix, and $a_{ij}^{(n)}$ is the element in row i and column j of A^n, the nth power of A, then $a_{ij}^{(n)}$ is the number of chains of length n from i to j; that is, we can determine the chains of length n between two points from the nth power of A. In counting by this method we include redundant chains as well as nonredundant ones. The inclusion of redundant chains stems from the symmetry of the adjacency matrix. For any two directly connected points, i and j, say, there is a chain of length 3: namely $i \to j \to i \to j$. In fact, there is a chain of length n, where n is any odd number, from i to j. This phenomenon on constructing chains by using the reverse ($j \to i$) of the relation just previously used ($i \to j$) we will call "doubling back."

Coleman presents a method which can be extended to prevent doubling back. He suggests zeroing out elements of the adjacency matrix as they are used; this zeroing out corresponds to removing a directed relation as seen as it is used to construct a chain, thereby preventing its reuse. We can prevent doubling back by zeroing out the symmetric relation as well as the one used to create the chain; that is, we zero our a_{ji} as well as a_{ij}.

Coleman's method can be defined as follows. We start from the $m \times m$ adjacency matrix A. Let $R_i^{(n)} = (r_{j;i}^{(n)})$ be a row vector of m elements; $r_{j;i}^{n}$ will be the number of chains of length n we count from i to j. Let $A_i^{(n)} = (a_{kj;i}^{(n)})$ be an $m \times m$ matrix; $A_i^{(n)}$ will result from the adjacency matrix A after the nth step from i. We define $R_i^{(n)}$ and $A_i^{(n)}$ inductively:

(1) $R_i^{(1)} = (r_{j;i}^{(1)})$; $r_{j;i}^{(1)} = a_{ij}$; that is, $R_i^{(1)}$ is the ith row of A;

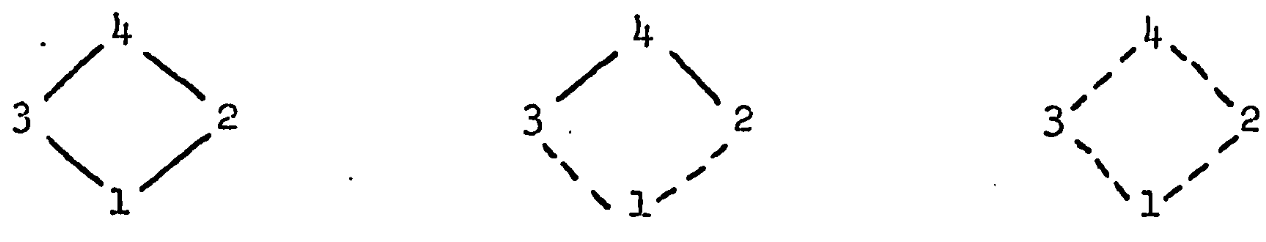
$A_i^{(1)} = (a_{kj;i}^{(1)})$; $a_{kj;i}^{(1)} = a_{kj}$ if $k,j \neq i$; $a_{ij;i}^{(1)} = 0$; and $a_{ki;i}^{(1)} = 0$; that is, $A_i^{(1)}$ is A with its ith row and column zeroed out.

(2) $R_i^{(n)} = R_i^{(n-1)} A_i^{(n-1)}$;

$A_i^{(n)} = (a_{kj;i}^{(n)})$; $a_{kj;i}^{(n)} = 0$ if $r_{k;i}^{(n-1)} a_{kj;i}^{(n-1)} > 0$ or $r_{j;i}^{(n-1)} a_{jk;i}^{(n-1)} > 0$;

and $a_{kj;i}^{(n)} = a_{kj;i}^{(n-1)}$ otherwise; that is, $A_i^{(n)}$ results from $A_i^{(n-1)}$ by zeroing out the elements corresponding to links used in constructing chains of length n from i.

This method, however, does distort out intuitive notions in some ways. The first way is its failure to find "the other path". Let us consider the simple structure in A of figure 9. Computing the chains from point 1, we notice that



A B C

Figure 9

there are two chains of length 1 from point 1: one from 1 to 2, and one from 1 to 3. B in figure 9 shows the structure we are left with after step 1; the broken lines indicate relations which have been zeroed out. At step 2, we find two chains of length 2: both from 1 to 4. We are now left without any remaining

relations, as shown in C of Figure 9. We failed to find the two chains of length 3: the one between 1 and 2 and the one between 1 and 3.

There is yet another way in which our intuition is distorted; this problem we will call "multiplication when paths join." Consider the structure in Figure 10. We note that our method would compute two chains, each of length 2, from 1 to 4; it would also compute two chains, one of length 1 and the other of length 2, from 2 to 4. While our method would find two chains from 1 to 5, it would only find one chain from 2 to 5, because the chains from 2 to 4 are of different lengths.

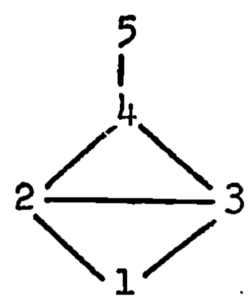


Figure 10

There is a way of partially correcting for this problem. We noted that $r_{j;i}^{(n)}$ is the number of chains of length n which we count from i to j; $r_{j;i}^{(n)}$ is an element in the row vector $R_i^{(n)}$ which is used to compute the chains of length n+1 from i. Let us define a row vector $Q_i^{(n)} = (q_{j;i}^{(n)})$ such that $q_{j;i}^{(n)} = 1$ if $r_{j;i}^{(n)} > 0$; that is the jth element of $Q_i^{(n)}$ is 1 if there are one or more chains from i to j of length n. Then we use $Q_i^{(n)}$ to compute the chains of length n+1 from i; that is, $R_i^{(n+1)} = Q_i^{(n)} A_i^{(n)}$.

Let us now define the similarity measure for any two points i and j. If the reader will remember, the similarity measure should be a function of the number of nonredundant chains between i and j and the lengths of these chains. $r_{j;i}^{(n)}$ is the number of nonredundant chains from i to j as best we can count them. To control for length we introduce an attenuation constant, a, where $0 < a < 1$. We then define $s_{ij}^{*} = \sum_{n=1}^{k} a^n r_{j;i}^{(n)}$, where K is the step at which the process is exhausted. Since s_{ij}^{*} is not necessarily equal to s_{ji}^{*}, we define the similarity measure s_{ij} as: $s_{ij} = s_{ji} = \dfrac{s_{ij}^{*} + s_{ji}^{*}}{2}$.

BIBLIOGRAPHY

Abelson, Robert P.

1966 "Mathematical models in social psychology," in Leonard Berkowitz (ed.) Advances in Experimental Social Psychology, Volume III, New York.

Agger, Robert, et.al.

1964 The Rules and the Ruled. New York, John Wiley & Sons, Inc.

Ball, G.H.

1965 "Data analysis in social sciences: what about the details?" Proceedings Fall Joint Computer Conference. Stanford Research Institute, Menlo Park, California, 533-559.

Coleman, J.S.

1961 The Adolescent Society. New York, Free Press.

Coleman, J.S.

1964 Introduction to Mathematical Sociology. New York, Free Press.

Coleman, James S. & MacRae, D., Jr.

1960 "Electronic processing of sociometric data for groups up to 1000 in size," American Sociological Review, 25, 722-727.

Crane, Diana

1969 "Social structure in a group of scientists: a test of the 'invisible college' hypothesis," American Sociological Review, 34, 335-352.

Green, P.E., Carmone, F.J. & Robinson, P.J.

1968 "Non-metric scaling methods: an exposition and overview," Wharton Quarterly, Winter-Spring, 28-41.

Guttsman, W.L.

1963 The British Political Elite. New York, Basic Books

Harary, Frank, Norman, Robert Z. & Cartwright, Dorwin

1965 Structural Models: an Introduction to the Theory of Directed Graphs. New York, John Wiley & Sons, Inc.

Hubbel, C.H.

1965 "An input-output approach to clique identification," Sociometry, 28, 377-399.

Kadushin, Charles

1968 "Power, influence and social circles: a new methodology for studying opinion makers," American Sociological Review, 33, 685-699.

27

Bibliography 2

Kruskal, J.B.

1964 "Multidimensional scaling by optimizing goodness of fit to nonmetric hypothesis," Psychometrika, 29, 1-27.

Luce, R.D. and Perry, A.D.

1949 "A method of matrix analysis of group structure," Psychometrika, 14, 95-116.

Menzel, Herbert, Coleman, James & Katz, Elihe

1966 Medical Innovation

Moreno, J.L.

(1934)
1953 Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Sociodrama. Rev. & Enlarged Ed. Beacon, N.Y., Beacon House.

Mullen, N.C.

1968 "The distribution of social properties in informal communication network among biological scientists," American Sociological Review, 33, 786-797.

Rubin, Gerold & Friedman, Herman P.

1967 A Cluster Analysis and Taxonomy System for Grouping and Classifying Data. New York, I.B.M.

Spilerman, S.

1966 "Structural analysis and the generation of sociograms," Behavioral Science, 11, 312-318.

Young, Ruth C. & Larson, Olaf

1965 "A new approach to community structure," American Psychological Review, 30, 926-934.

The Sociogram System

A User's Manual

Richard Alba

June 1970

Teachers College

and

Bureau of Applied Social Research,

Columbia University

Users Manual for the Sociogram System

The General Flow of the System

The first step in the procedure is to partition the population into disjoint subsets, each one of which will become a sociogram.

The population is renumbered by RENUM, yielding a set of observations whose id's are numbered sequentially from 1. The renumbered population is then input to the connectivity program (CONNCT2) which computes a relatedness matrix and an adjacency matrix. Either of these is then input to the Hierarchical Clustering program (HIER), from whose output the disjoint subsets may be read.

A particular subset is then renumbered once again and relatedness and adjacency matrices are computed. The relatedness matrix is then scaled by MDSCAL into two dimensional space and the optimum configuration is punched out. This configuration together with the adjacency matrix is then input to the plotting program (PLOT) which produces an output which may be plotted on the 4060.

The Renumbering Program (RENUM) -- version 3

The purpose of the renumbering program is to renumber a body of sociometric data so that the respondent id's proceed in sequence, starting from 1.

The program allows the user to delete individuals from the sample or add to the sample individuals who are named by someone in the sample but are not themselves represented by an observation in the sample. In the case where individuals are deleted from the sample, all references to these individuals by others in the sample are deleted. Individuals are deleted from the sample because their id appears on a list, called the acceptance/rejection list, supplied by the user (or fails to appear) or because they are named too few times by others in the sample (the number of times they must be named to be retained is a parameter to the program).

The third version of the renumbering program allows jobs to be batched; that is, the user may renumber as many different bodies of sociometric data as he wishes in one submission.

<u>control cards</u>

1. the parameter card

    cols 1-5: NE -- the exact number of people on the list of those to be accepted or rejected; the parameter REJ determines whether these people will be accepted or rejected; leave this field black or code a 0 if there is no such list;

    cols 6-9: NC -- the maximum number of choices per observation;

cols 10-13: KONS -- the number of times an individual presently represented by an observation in the sample must be chosen to be retained in the sample;

cols 14-17: KONS1 -- the number of times an individual named but outside the sample must be chosen to be represented in the sample; note that if this field is non-zero and an individual outside the sample fails to be named a sufficient number of times, then all references to this individual will be deleted in the output; that if this field is zero then all individuals who are named but outside the sample are represented by observations in the output; that if this field is non-zero and an individual outside the sample is named a sufficient number of times, then this individual is represented by an observation in the output.

If this parameter is greater than the number of people in the sample, then all references to individuals outside the sample will be deleted in the output.

col 18: REJ -- code a 0 if the people on the acceptance/ rejection list are to be rejected; code a 1 if anyone not on the list is to be rejected; if there is no list (NE is 0), then the contents of this field are not used;

col 19: SMPL -- code a 0 if there is to be no simplication through the elimination of duplicate choices; code a 1 if simplification is to take place; this simplification occurs when one person is chosen twice by another--in this case, one choice is eliminated.

2. the input format card

This card contains a FORTRAN format statement which is used to read the data records. Only integer format items should be used. The first item in the format list corresponds to the respondent's id. All subsequent items correspond to the choices of the respondent.

The first item of the format statement is also used to read the acceptance/rejection list if there is any; that is to say, each record in the acceptance/rejection list contains one item of the list and that item is located in the same position in the record as the respondent's id is located in the data record.

3. the output format card

This card contains a FORTRAN format statement which is used to write the output records; these records are written on the file whose ddname is FT07F001 and are, thus, normally punched. Each item in this format statement is an integer item. Each output record will contain the following information: the old identification number of the respondent, the new renumbered identification number, and the renumbered identification numbers of this individual's choices, if any.

4. the title card

The title card may contain any alphameric information.

the deck setup

The order of control cards to process any deck of sociometric data is as follows:

title card

input format

output format

parameter card

acceptance/rejection list, if NE is non-zero

data.

At Columbia, the following is the job setup which is required to run the program:

```
/*SETUP  DEVICE=2311,ID=MDKØØ4
//JOBLIB  DD  DSN=SYS5.SOCIO,DISP=(SHR,PASS),
//  UNIT=2311,VOL=SER=MDKØØ4
//A  EXEC PGM=RENUM,TIME=n
//FTØ6FØØ1  DD  SYSOUT=A
//FTØ7FØØ1  DD  SYSOUT=B
//FTØ5FØØ1  DD *
     control cards and data
/*
```

multiple submission

The order of control cards and data for the processing of multiple decks in one job submission is as follows:

COLEMAN DATA -- AS USUAL 1 X 1

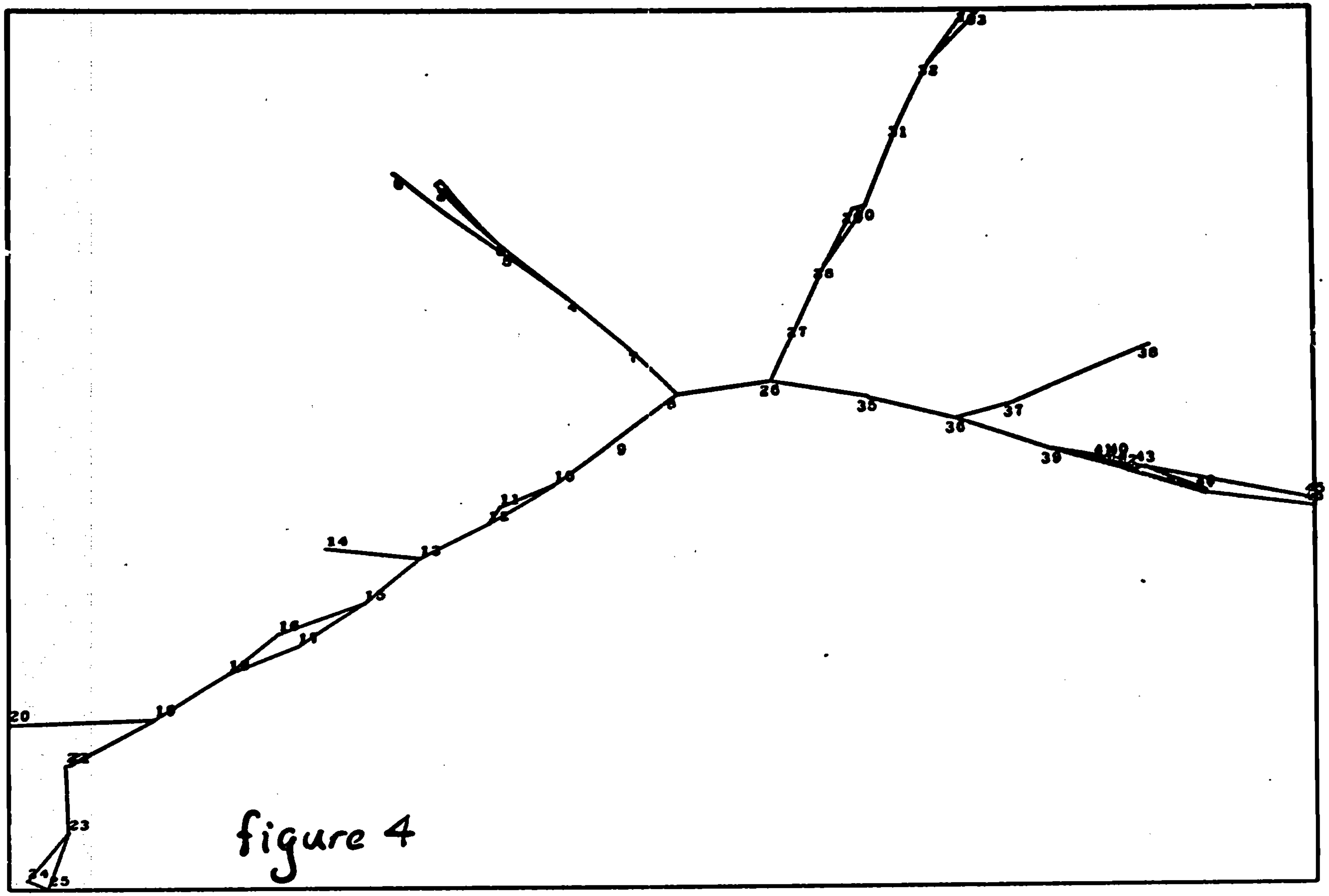


figure 4

YUGO (SECTOR 1) -- V2 (A=.5) -- SECONDARY, CUTOFF=.00001, S=.38 -- 1 X 1

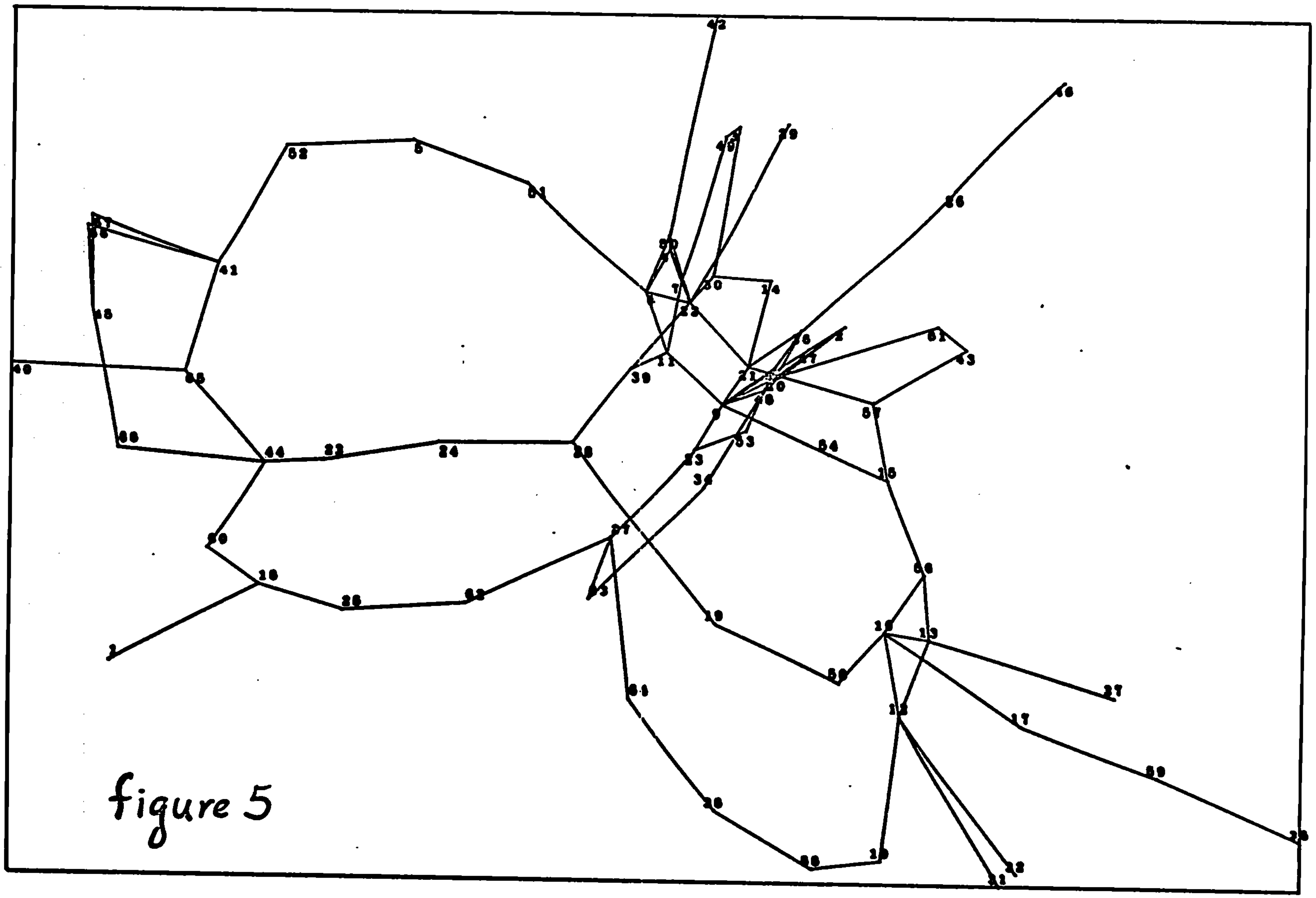


figure 5

YUGO (SECTOR 3) -- V2 (A=.5) -- SEC,CUT=.00001,S=.38 -- 1 X 1

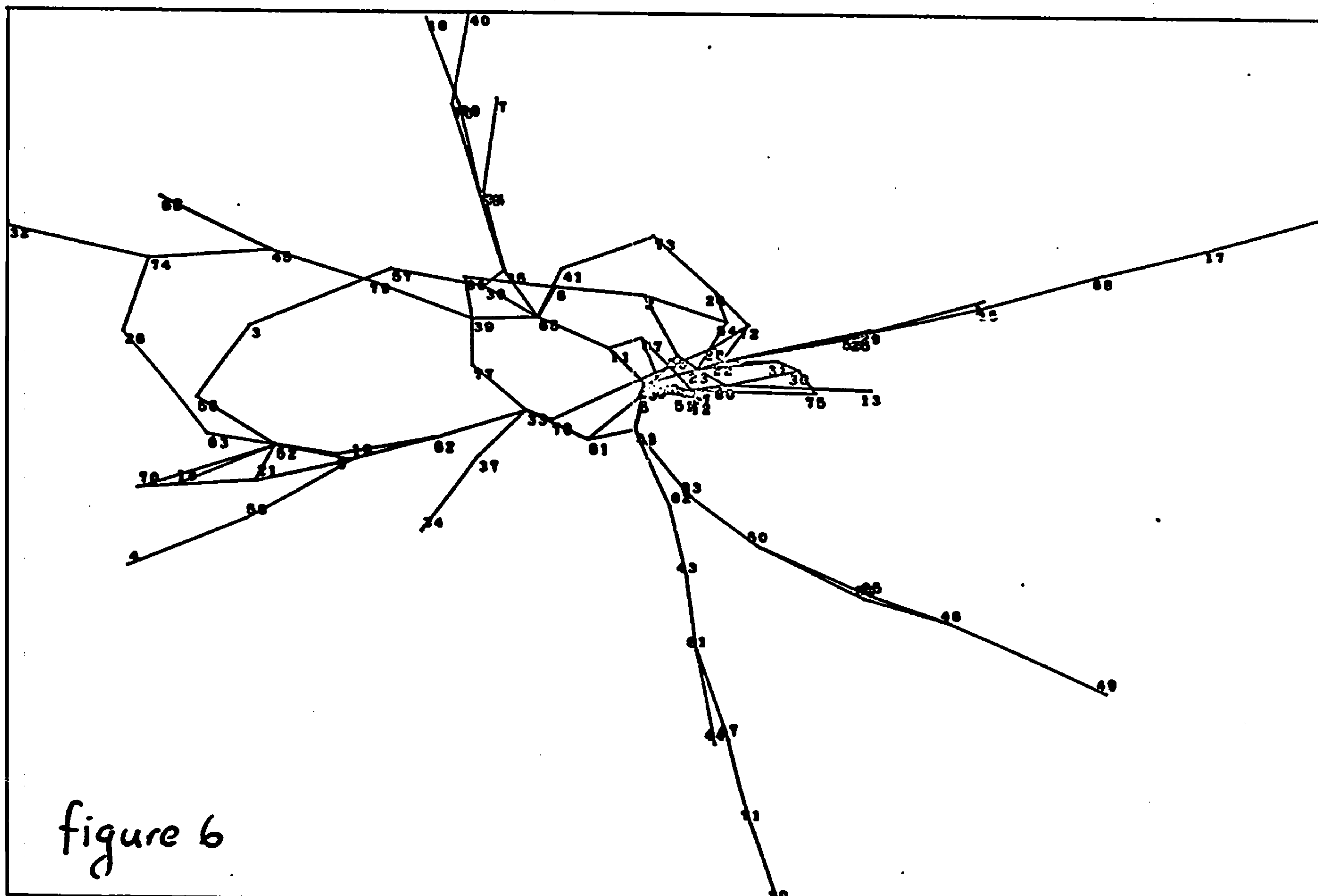


figure 6

YUDD 3 -- V2 (A=.5) -- SEC,CUT=.00001,S=.35 -- CUT NEAR & PAR -- 1 X 1

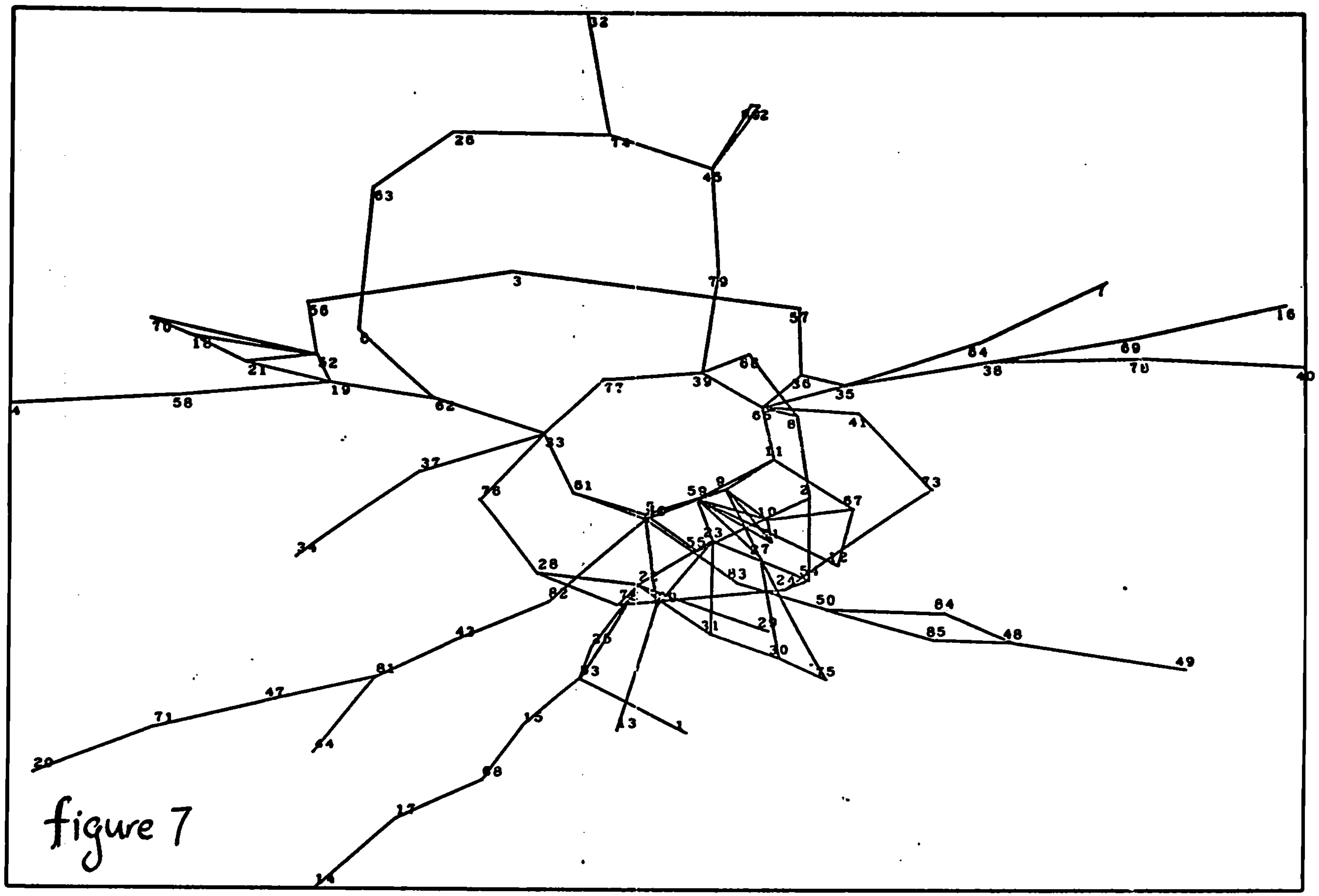


figure 7

job 1 = control cards and data

blank observation

job 2 = control cards and data

blank observation

.
.
.

job n-1 = control cards and data

blank observation

job n

The output decks are separated, in a multiple deck submission, by a separator card in which punches 12,1, and 9 occur in every column.

restrictions

1. Before any deletion, the total number of individuals in the sample plus the total number of individuals named but not in the sample must not exceed 3000.

2. NC must not exceed 15. If NC exceeds 9, however, the printed output may not be correct (although the punched output will be correct).

Connectivity Program

There are two versions of the connectivity program. The first version counts all paths; the second counts distinct paths. As a general rule, the second is probably the more useful.

The parameter cards for each version are the same and follow:

(1) the card which specifies the number of individuals; it is of the form:

N = int, where int is an integer.

for example:

N = 87

(2) the card which specifies the attenuation constant; it is of the form:

ALPHA = real, where real is a real number.

for example:

ALPHA = .50000

(3) the card which specifies the maximum number of iterations to be performed; it is of the form:

OMEGA = real, where real is a real number.

for example:

OMEGA = .0002

the cycle n for which $(\text{ALPHA})^{n}$ OMEGA but $(\text{ALPHA})^{n+1}$ OMEGA is the last to be computed.

(4) the card which specifies the maximum number of choice; it is of the form:

NC = int, where int is an integer.

for example:

NC = 3

These cards appear in the order in which they are stated here. Following them are two format cards, the first of which specifies the input format, and the second of which specifies the output format.

The only fields for which provision is made in the input format are those which identify choices of the respondent. Each of these is designated by an 'I' format specification.

For example, suppose that there are five two digit fields on a card in columns 1 through 10; that the first is the recoded identification of the chooser, the second is the original identification of the chooser, the third through the fifth are the recoded identifications of the chosen. Then the input format would appear as follows:

(4X, 3I2)

In addition to printed output, there are two putput data sets. A connection matrix, a matrix of Ø's and 1's, where 1 indicates a first-level connection, is written on the ddname FT08F001. The format statement associated with this data set is (25F3.Ø). No identification number appears in the output record. It is suggested that this data set always be stored on disk or tape.

A connectivity matrix, a matrix of positive real numbers, where relative magnitude indicates the degree of relatedness, is written on the ddname FT07F001. It is this data set for which the user must provide an output format statement. Again, no identification number appears on the output record, and so it is suggested that this data set always be stored on disk or tape.

To summarize, then, the control cards appear as follows in a job setup:

N = int

ALPHA = real

OMEGA = real

NC = int

input format

output format

data, which is the output from the renumbering program

To use the connectivity program at Columbia, the following job setup is required:

```
/*SETUP  DEVICE=2311,ID=MDKØØ4
/*SETUP  DEVICE=2311,ID=DCUØ32
//JOBLIB  DD DSN=SYS1.MAT4ØØ,DISP=(SHR,PASS),
//  UNIT=2311,VOL=SER=DCUØ32
//A  EXEC PGM=DFRMTL,TIME=n,PARM='DCALNG=m'
//FTØ3FØØ1  DD  SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//MTLPROG  DD  DSN=SYS5.SOCIO(CONNCT2),DISP=(SHR,PASS),
//  UNIT=2311,VOL=SER=MDKØØ4
//FTØ6ØØ1  DD  SYSOUT=A
//FTØ7FØØ1  DD
                              output data set
//FTØ8FØØ1  DD
//FTØ1FØØ1  DD  *
      control cards and data
/*
```

m, the DCALNG parameter, is calculated as follows:

$m = (2\emptyset * N^2)/(1\emptyset\emptyset\emptyset)$, where N is the number of individuals.

Moreover, the REGION parameter on the JOB card is approximately 45Ø+m.

<u>restriction</u>

1. The maximum number of choices, NC, may not exceed 20.

The Scaling Program (MDSCAL)

The scaling program (MDSCAL) was developed by Dr. J. B. Kruskal of Bell Laboratories and F. J. Carmone.

strategy

We use the scaling program to find the configuration of points in 2-dimensional space whose distances best fit the relatedness or connectivities. The stress measure which is printed by the program is a measure of goodness of fit; we seek that configuration for which stress is a minimum.

To find this particular configuration it is best to repeat the scaling several times, each time starting from a random configuration. In our own use of the system, we usually perform five scalings of the data, selecting for plotting the one producing the minimum stress. It should be noted that as many scalings as desired may be performed in one job submission.

There are normally two job submissions involved in one use of MDSCAL: one to locate the best configuration and one to obtain a deck containing it. To obtain a deck it is only necessary to place the CARDS card in the MDSCAL job (see below) and rerun the job without further change. It should be noted that no changes may be made in the jobs preceding the one producing the minimum stress if it is to produce the same output.

MDSCAL control cards

The following cards, punched as shown, constitute an MDSCAL

job. With exceptions as noted, the statements may be punched starting in any column.

DIMMAX=2,DIMMIN=2

CARDS, if the configuration is desired on cards

SECONDARY

RANDOM=n,n any integar

ITERATIONS=1ØØ

CUTOFF=.ØØØØØØ1

DATA REGRESSICN=DESCENDING

title card, title in columns 1-80

parameter card: cols. 1-3 contain N, the number of points; col. 6 contains 1; col. 9 contains 1.

format statement corresponding to the input data, which is the relatedness matrix

COMPUTE

<u>job setup</u>

```
/*SETUP tape containing the relatedness matrix.

/*SETUP DEVICE=2311,ID=MDKØØ4

//JOBLIB DD DSN=SYS1.MDSCAL,DISP=(SHR,PASS),UNIT=2311

// VOL=SER=MDKØØ4

// EXEC PGM=MDSCALTA,TIME=n

//FTØ2FØØ1 DD UNIT=DISK,SPACE=(80,(25))

//FTØGFØØ1 DD SYSOUT=A

//FTØ7FØØ1 DD SYSOUT=B

//FTØ8FØØ1 DD data set containing relatedness matrix

//FTØ5FØØ1 DD *
```

```
MDSCAL job 1
  .
  .
  .
MDSCAL job N

STOP

/*
```

The REGION parameter is 400K.

Limitations:

1. N may not exceed 100.

PLOT

Plot takes the configuration produced by MDSCAL together with adjacency matrix and produces from them the plot of the associated graph structure. This plot is output by the program in the form of a data set on tape which can be plotted by the META processor on the 4060, the cathode ray tube plotter.

PLOT plots each point according to its coordinates in the configuration and draws lines connecting points where they are indicated by the adjacency matrix.

For legibility, the plot of the sociogram may be blown up over a picture which is several pages long as well as the same number of pages wide. Additionally, titles may be provided for each page to keep track of the output.

<u>input data sets</u>

There are three input data sets to PLOT, besides the data set of control cards, FT05F001: the data set containing the configuration produced by MDSCAL, FT08F001; the data set containing the dictionary of identifications to be associated with the points, FT09F001; the data set containing the adjacency matrix, FT10F001.

1) the configuration produced by MDSCAL.

This data set is normally on cards. The first four cards are removed from the data set before inputting it to PLOT.

The format statement corresponding to this data set should contain only two fields, each one corresponding to the coordinate of one dimension. If the points were scaled into a space higher than

two dimensions, only two may be selected for plotting. Each item in the format statement should be an F or E item.

2) the dictionary of point identifications.

This data set is normally on cards. It may be the data set containing the renumbered identifications produced by RENUM, the renumbering program.

Each record normally contains two numeric fields: the first is the identification number to appear in the output; the second is the present (sequential) identification. These fields occur in the order stated. An I format item (in the formal statement) corresponds to each field.

This data set is used in the following manner. If the data set is empty, each point is identified in the output by its sequential identification. Otherwise, the sequential identification of each point which has a corresponding record in the dictionary data set is replaced by the value of the first field of the record.

Note: if this data set is empty, there must still be a corresponding format and DD statement.

3) the adjacency or connection matrix.

This data set is normally on tape. It is produced by the connectivity program.

This matrix is an N x N matrix, where N is the number of points. Each entry in this matrix is a 0 or 1; 0 indicates the absence of a connection and 1 the presence.

The format statement corresponding to this data set is normally (25F3.0).

control cards

Note: the control cards appear in the deck in the order in which they appear below.

1) the title card. This card contains information which will be placed at the top of each page.

2) the jobtitle card. The information on this card appears on the first page of the 4060 output, which serves as a burst page. The fields on this card are:

cols. 1-8. the project number, i.e., the CUCC account number.

cols. 9-16. the programmer name.

cols. 17-24. the jobname. any title which is meaningful to the user.

3) the format statement for the configuration from MDSCAL. This format statement, as well as all others, must be punched on one card.

4) the format statement for the adjacency matrix. Normally (25F3.0).

5) the format statement for the dictionary.

6) the parameter card. The fields are:

cols. 1-3. the number of points in the graph.

cols. 4-6. the length of the graph, i.e., the number of pages long and wide.

job setup

At CUCC, the following job setup may be used:

/*SETUP tape to contain 4060 output

/*SETUP tape containing adjacency matrix

/*SETUP DEVICE=2311,ID=MDKØØ4

```
//JOBLIB DD DSN=SYS5.SOCIO,DISP=(SHR,PASS),UNIT=2311,VOL=SER=MDKØØ4

// EXEC PGM=PLOT,TIME=n

//FTØ6FØØ1 DD SYSOUT=A                 (Ø indicates a digit zero
                                        O indicates a letter O)

//FT1ØFØØ1 DD data set containing adjacency matrix

//SC4Ø6ØZZ DD data set containing 4060 output

//FTØ5FØØ1 DD *

  control cards

/*

//FTØ8FØØ1 DD *

  output configuration from MDSCAL

/*

//FTØ9FØØ1 DD *

  dictionary

/*
```

<u>limitations</u>

1. The number of points may not exceed 1000.
2. The number of connections or relationships may not exceed 3000.
3. The number of digits in an identification may not exceed 4.

Hierarchical Clustering (HIER)

The hierarchical clustering program was developed and written by S. C. Johnson of Bell Labs.

strategy

The primary use for the hierarchical clustering program is to separate a population into disjoint subsets. This procedure is as follows. The population is renumbered and a relatedness matrix is computed for it. This relatedness matrix is then clustered using HIER. Disjoint subsets can be discovered in the printed output labeled "Connectedness Method." Each subset will be a contiguous set of identifications in the printed output which is connected at the 0.0 level to every individual not in the subset.

Additionally the "Diameter Method" may be used for detailed sociometric analysis.

control cards

The control cards are as follows; they appear in the deck in the order in which their descriptions occur below:

a) Title card. Cols. 1-80 may contain any text.

b) Parameter card. The fields are as follows:

cols. 1-3. The number of individuals to be clustered.

cols. 4-5. Punch a -1.

col. 6. Punch a 1 if the data (relatedness matrix) is on tape. Otherwise, leave blank. If the data is on tape an FT08F001 DD statement must appear.

cols. 7-16. A missing data code. The value punched here will replace any zero in the relatedness matrix. Leave blank if the zeroes in the relatedness matrix are to remain unchanged (the normal situation).

c) The format card by which the relatedness matrix is to be read.

data

The data is normally the relatedness matrix produced by the connectivity program. The matrix may be either on cards, in which case it is placed right after the control cards in the deck, or on tape, in which case there must be an FT08F001 DD statement describing it.

job setup

At CUCC the job setup is as follows:

```
/*SETUP tape containing relatedness matrix
/*SETUP DEVICE=2311,ID=MDKØØ4
//JOBLIB DD DSN=SYS5.SOCIO,DISP=(SHR,PASS),
// UNIT=2311,VOL=SER=MDKØØ4
// EXEC PGM=HIER,TIME=n
//FTØ1FØØ1 DD UNIT=DISK,SPACE=(3624,(m)),
// DCB=(RECFM=V,LRECL=362Ø,BLKSIZE=3624)
//FTØ6FØØ1 DD SYSOUT=A
//FTØ8FØØ1 DD description of the data set containing the
                relatedness matrix.
//FTØ5FØØ1 DD *
      control cards and, possibly, data
/*
```

The region parameter on the job card must be calculated as $52 + \text{buffers} + \frac{2N(N-1)}{1000}$, where N is the number of individuals to be clustered and 'buffers' is the space for input-output buffers (as a general rule, 15K will be adequate).

The parameter 'm' in the FT01F001 DD statement is the next integer larger than $\frac{2N(N-1)}{3620}$.

CHAIN

User's Manual for a Sociometric Linkage Program

Richard Rosen

Bureau of Applied Social Research
Columbia University

and

Peter A. Abrams

Bureau of Applied Social Research
Columbia University

January, 1970

The development and programming of CHAIN was supported by grants from the Office of Education, U.S. Department of Health, Education and Welfare.

Abstract

CHAIN is a program which aids in the analysis of sociometric data. In its present stage the program analyzes the relationships between any two elements of a group by constructing all possible linkages, or paths, between the two. The result is much the same as that produced by algorithms involving matrix multiplication but execution time is much faster. Further, unlike matrix multiplication algorithms, most of which have no end point, CHAIN terminates construction of a chain for a given element when no new elements are added in expanding the sociometric choices. Various matrices, distributions and statistical measures are produced during execution of the program.

CHAIN is written in Fortran IV for and IBM-360 with at least 250K bytes of storage and is designed to process up to 999 individuals making a maximum of 19 choices per person. The capacity of the program may be increased by enlarging the dimensions as long as appropriate storage is available at your installation.

Description

Basic output:

This program can be used to obtain the following measures for a group and the elements or individuals of the group:

1. a distance matrix indicating the shortest path from any person to any other person
2. an ultimate connectedness matrix indicating if two people are in any way connected to each other
3. linkages for each individual at each remove
4. aggregate measures characterizing the choice network for each individual as well as the network of choices received by that person, termed forward and backward connectedness, respectively
4. a distribution of the number of choices (and times chosen) and summary statistics for each person by remove
5. Coleman's measure of ultimate connectedness (percent of possible connections)
6. summary statistics of both the rows (choices) and columns (times chosen) of the entire distance matrix

Optional features:

CHAIN can also be used to perform operations on the input data before analysis. The following options may be performed:

1. deletion of non-reciprocated choices
2. addition of non-reciprocated choices
3. punch out a reformatted input deck, i.e., if the input deck contains duplicate choices and embedded blanks or zeros the CHAIN program will eliminate them, left adjusting the choices and output a new deck

Algorithm:

Given person i and his k sociometric selection, $p_{i1},\ldots,p_{ik}$, a chain is generated in the following manner:

a. consider the k choices of person i as direct choices (1 remove)

b. look at the k choices of each of the previous k chosen persons; if they are represented at the first remove then we ignore them, if not they are considered to be 2 removes from i

c. repeat step b for the new choices generated until there are no longer any new choices in i's chain; the new choices at the i-th cycle represent choices i+1 removes away from person i

d. record the constructed chain and move to the next person

e. repeat until all persons in the group have been analyzed producing the distance matrix of shortest paths from person i to j. Summary statistics are produced from this matrix

Coding of the data:

Each individual responding to the sociometric item and each person selected by him must be given unique integer identification number. The input data deck must contain a card(s) for each person numbered even if he makes no choices. That card(s) would contain only an ID number in the appropriate columns.

The data card(s) for each person in the group to be analyzed must contain the following information in the following sequence:

a. a previous or old ID number, if no such number exists, i.e., if the deck is already serially numbered, the user must leave several columns blank at the beginning of the card. This field is used by

the program in punching statistical output so that it may be correlated with other data available for the subjects

b. the second field should be the person's sequential ID number and the data should be input in this sequence 1,...,n

c. the remaining fields are the ID numbers of the people he chooses, recoded according to the sequential list

The number of cards per person may vary since the user indicates to the program the format of the input data. Again, all persons as input to CHAIN must be numbered serially. The largest ID number must correspond to the number of people as input. A choice of person i who makes no choices himself must have his own input card.

Organization of the data sets:

Multiple groups are allowed, the entire CHAIN routine being repeated for each group. The first card of any data set (here we mean the entire data input to CHAIN including control cards) indicates the number of groups which the program is to analyze. the number is punched right justified in columns 1 to 4. This card is used only once and preceeds all groups and their control cards. The data input for each group follows this data set card but each group must conform to the following organization:

1. Title card
2. Control card(s)
3. Format card(s)
4. Data Cards

1. Title card: Alpha-numeric information punched in columns 1-80 of this card will be used to label the output. If no label is desired a blank card must be inserted. Blanks may occur in the title card.

2. Control card(s): (all numbers are punched right justified)

CARD 1	Column(s)	Description
	3- 5	Number of people in the group to be analyzed, may vary to 999 but may not exceed that figure, must be punched on card
	9-10	Number of sociometric choices per person, may vary to 19 but cannot exceed this figure, must be punched on card
	15	2=data is to be symmetricized by adding non-reciprocated choices 3=data is to be symmetricized by deleting non-reciprocated choices 0,BLANK=neither of the above is desired, raw data used
	20	1=new deck is desired after reformating has been done 0,BLANK=new deck is not desired
	25	1=each person's chains are to be written out (Warning:this option can produce excessive amounts of printed output depending on the size of the group and its density) 0,BLANK=chains not desired
	30	1=user wants connectedness matrix (i,j-th cell is printed "X" if i and j are ultimately connected, "." if not) 0,BLANK=connectedness not desired
	35	1=user wants distance matrix 0,BLANK=distance matrix not desired

Column(s)	Description
40	1=all rows of matrix are to be outputted before next column 0,BLANK-matrix form of output for either connectedness or distance matrix desired
41-45	Length of longest indirect choice chain to be used, integer, right justified, should not be larger than number of people in group BLANK=default value of N, the size of the group
50	1=punched card or written (on users tape) statistical output desired; old ID is written before sequential ID used by program 0, BLANK=statistical data not desired
55	Indicates whether or not user desires to change I/O logical unit numbers for use during execution (with the exception of the card reader=5 and printer=6 which must be changed in the source deck) 1=some or all optional device numbers are to be changed 0,BLANK=default values are to be used If user indicates a change is desired in one or more of the optional unit device numbers he <u>must</u> insert CARD 2, if not, this second card is omitted
56-65	Attenuation constant used in statistical measures; digits to the right of column 60 assume fractional value, no decimal point is needed; 0,BLANK=default value of 0.5000 NOTE: unless there exist good reason for not doing so, it should be fractional

CARD 2 *	Column(s)	Description
*(optional device card-not to be inserted if column 55 of first control card is blank)		
	3- 4	Scratch tape or disk for recording connectedness matrix ** DEFAULT VALUE= 1
	7- 8	Scratch tape or disk for recording distance matrix ** DEFAULT VALUE= 2
	11-12	Input device for data DEFAULT VALUE= 5 card reader
	15-16	Statistical output device (if column 45 of card one is a 1) DEFAULT VALUE= 7 card punch
	19-20	Output device for reformated deck DEFAULT VALUE= 7 card punch

** The data on these units is written out in an unformatted mode. The JCL for these units should be variable blocked.

The amount of data written on these units varies with the number of people being analyzed. This fact is important when disks are used as scratch units. For less than 200 people with 5 choices per person, SPACE=(3620,(100,100)) , should suffice. The most space used was for 900 people with 15 choices (this group was very dense) and amounted to 400 tracks, i.e., SPACE=(3620,(400,100))

3. Format Card(s): Only one format card is required if no new deck is to be punched. This format card indicates the arrangement of the data to be inputed. Columns 1-80, left justified, on this card may be used. The first field specified must be for the old ID number, the second for the serialized ID number that is used by this program for processing, the remaining fields represent the person's choices. All fields must be integer.

Example 1: No old Id is available, serial ID is in columns 3-5, five choices, each in 4 columns beginning in column 8

(I2,I3,2X,5I4)

Example 2: Old ID in col 1-4, new ID in col 8-10 and 14 choices nine of 3 columns and five of 4 columns starting in column 11

(I4,3X,10I3/5I4)

Second format card: used only if a new deck option has been indicated on the program control card. This format card specifies the way in which the user wants the reformated deck punched out. The program does not output an old ID number when punching a new deck. The number of outputed items is, therefore, one less than that inputed.

4. Input data: Data may be on cards read in from tape or disk. If either of the latter two options are desired the correct column must be punched 1 on the program control card and the correct device must be indicated on the second control card. Data must conform to the requirements set earlier and must be in the correct order. If not, the program will detect an error and cancel; all suceeding program decks will be flushed.

References:

Coleman, James, Introduction to Mathematical Sociology, Free Press, 1964, Chapter 14

A Note on Further Use of the Distance and Connectedness Matrices

By specifying the optional tape or disk devices the user can record the distance and/or connectedness matrix for further analysis. The method for indicating the use of user tapes or disks rather than scratch disks is described in the section on organization of data sets. However, it is necessary that the user be familiar with the format of this output in order to utilize it correctly.

Both the distance and connectedness matrices are written without the use of format control, i.e., in binary mode. In order to read them the user must use, if he is using a fortran program, an unformatted read statement. The use of such statements is described in the IBM publication, "Programmers Guide to Fortran."

Distance Matrix:

The format of the distance matrix is as follows: each row is written out serially. The first entry of a record is the row number followed by a vector of length N, where N is the number of persons in the group. Each entry in this vector represents a value from 1 to N (or if the user has specified a maximum chain length, a distance from 1 to the maximum), which is the distance that person i (if we are on the ith row) is from person j (the jth cell of the row vector). Hence, the user must first read the row number and then a vector of length N.

```
      REWIND 8
      DO 100 K=1,N
      READ(8) NROW,(NARRAY(I),I=1,N)
         .
        (user statements)
         .
  100 CONTINUE
```

In the example above, the data is on logical unit 8, NROW is the row number (chooser) and NARRAY(I) is the distance of the other persons from person NROW.

Connectedness matrix:

The connectedness matrix, on logical unit 1, unless the user specifies another device number, is formated as follows: Record A - the number of choices of the ith person (all removes) = NTOT; Record B - the number of the ith person and NTOT entries, each being an ID number of a person who is chosen by person I.

```
          REWIND 10
          DO 100 K=1,N
          READ(10) NTOT
          READ(10) I,(NARRAY(NK),NK=1,NTOT)
              :
              (user statements)
              :
      100 CONTINUE
```

In the example above, each entry in NARRAY will be an ID number (there are NTOT such entries) and each number identifies a choice of person I. The ID's are the sequential or new ID's.

JOB CONTROL LANGUAGE (JCL)

Example 1: Source deck is used. Input data on card reader. Scratch units not varied.

```
//JOB ...      ... ,REGION=300K
/*FORMAT PR,DDNAME=FT06F001,OVFL=ON
// EXEC FORTGCLG,PARM.FORT ...   ... ,TIME.GO=nn
//FORT.SYSIN DD *
          :
     (Source deck)
          :
/*
//GO.FT01F001 DD SPACE=(3620,(100,100))
//GO.FT02F001 DD SPACE=(3620,(100,100))
//GO.SYSIN DD *
          :
     (Data set(s))
          :
/*
```

Example 2: Object deck, input data on logical unit 8 (tape), scratch unit 1 (connectedness matrix), on logical unit 11 (tape), scratch unit 2 (distance matrix) on logical unit 12 (tape)

```
//JOB ...      ... ,REGION=300K
/*FORMAT PR,DDNAME=FT06F001,OVFL=ON
/*SETUP...
/*SETUP...(RING on units 11 and 12, NORING on 8)
/*SETUP...
// EXEC FORTGLG, ... ...
//LKED.SYSIN DD *
          :
     (object deck)
          :
/*
//GO.FT08F001 DD UNIT=2400-9,DISP=OLD,VOL=SER=xxxxxx,
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=80),DSN=INPUT,
//    LABEL=(n,SL)
//GO.FT11F001 DD UNIT=2400-9,DISP=NEW,VOL=SER=xxxxxx,
//    DCB=(RECFM=V,BLKSIZE=3620),DSN=CONNECT,LABEL=(n,SL)
//GO.FT12F001 DD UNIT=2400-9,DISP=NEW,VOL=SER=xxxxxx,
//    DCB=(RECFM=V,BLKSIZE=3620),DSN=DISTAN,LABEL=(n,SL)
//GO.SYSIN DD *
          :
     (data set(s))
          :
/*
```