THE APPLICATION OF CLUSTER ANALYSIS ON A POST OFFICE SCHEDULING PROBLEM

BY

SIU-SIK WONG

B.A.Sc., UNIVERSITY OF BRITISH COLUMBIA, 1972

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER IN BUSINESS ADMINISTRATION

in the Faculty

of

Commerce and Business Administration

We accept this thesis as conforming to the
required standard

THE UNIVERSITY OF BRITISH COLUMBIA

July, 1976

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Commerce and Business Administration

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date July 26, 1976

ABSTRACT

The application of computerized clustering methods in outlining the truck route boundaries for street letter box collection runs is believed to be an effective tool for use by the Vancouver Post Office. This study investigates and analyses the characteristics, algorithms, and applicability of 12 cluster analysis techniques in grouping sets of two-dimensional data units for the Post Office. A broad view of cluster analysis is presented, including a review of the methodology and the potential problems associated with nine hierarchical and three nonhierarchical clustering methods. Two sets of contrived data and two empirical data sets (consisting of street letter box locations in the Burnaby area) are used to test the suitability of the grouping methods in clustering both evenly and unevenly distributed data units in a 2-dimensional Cartesian space. Computer programs for various clustering procedures are used to generate tree diagrams showing the linkages of the members within each group as well as the membership lists for the four data sets. The results are then plotted onto maps for evaluation. Results of the evaluations, based on group sizes, distributions of distances within groups, and travel times and distances, can be summarized as follows:

a. Ward's method and the three nonhierarchical methods are better clustering techniques in grouping evenly distributed data sets;
b. the complete linkage method, and the two average linkage methods are more suitable for grouping visually identifiable clustered data units;
c. the single linkage methods and the centroid methods are generally less satisfactory in grouping all four sets of data; and
d. clustering techniques provide a useful tool for outlining the route boundaries for street letter box collections.

A comparative study for the Vancouver area would substantiate the feasibility of cluster analysis as an aid to solving the scheduling problem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

I wish to thank Dr. C.L. Doll, my committee Chairman, for his guidance and time during all stages of my thesis work.

I would also like to thank Mr. B. Dyke and Mr. T. Lim, both of Vancouver Post Office, for their assistance in collecting data.

And special thanks to my wife Charmaine for her assistance in the typing of this thesis.

CHAPTER I

INTRODUCTION

The Vancouver City Postal Transportation Services, like many postal services of other big cities, is constantly faced with complicated revisions of schedules in the pursuit of assigning suitable duties to various trucks and drivers. The scheduling of delivery services is highly constrained by the service and process requirements of the Post Office. The scheduling is further complicated by the unevenly distributed destinations of the processed mail volume. Planning for the truck and human resources movements is complex and time-consuming. At the present stage, the planning of all the schedules rely heavily on the experience of the experts and the management is currently encouraging the development of a more efficient scheduling technique.

One of the biggest problem in scheduling is the setting up of the boundaries in which each scheduled service would render its delivery effort. Clustering analysis technique is a suitable approach to solve the problem. With the advent of computerization, several cluster methods can be run in a single program and then their results can be aids to the planners and schedulers in the determination of the service boundaries. In order to apply this technique

effectively, the complex operation of the Vancouver City Postal Transportation Service must be comprehended.

1.1 Vancouver City Postal Transportation Service

The Vancouver City Transportation operation is a very complex system with a fleet of various sized vehicles performing different types of delivery and collection services. As of May 1976, the Post Office City Transportation system has 151 small vehicles of which 139 are scheduled for daily services, and 12 are kept on standby basis to meet contingencies, irregularies, replacements and breakdowns. A total of 23 medium to large trucks are also utilized for other transportation services within Greater Vancouver area.

Small vehicles of ½ ton capacities are mainly used for daily street letter box collections, relay bundle deliveries, parcel post deliveries, and special deliveries. Larger trucks are assigned to shuttle services transporting bulk mail volumes to and from airport, docks, railways stations, satellite post offices and the Vancouver General Post Office. The scheduling of these different sized trucks for various services is very intricate and complex. Each type of service has its own characteristics in timing, location constraints, and degree of importance to the Post Office operations. Frequent rescheduling of services are required because there are often alterations in the airplane

schedules, delivery requirements, services requirements, street letter box and bundle box allocations, mail volume pattern, and union regulation. The rescheduling of truck services is presently done manually by the planners.

Constant growth of mail volumes to and from the Greater Vancouver area in recent years is brought about by the tremendous increases in population as well as small industries in the area. The coordination of city postal transportation is becoming a very intricate and difficult problem to tackle. A special vehicle utilization study in the summer of 1975 investigating the feasibility of decentralization of North and West Vancouver Mail Services from the General Post Office indicated that efficiency of the system could be highly improved by decentralization.[1] The study also introduced computerization in volume statistics and routing of vehicles. These computerized methods, undoubtly, help the planners and the management in the scheduling of services, however, they cannot be utilized efficiently without the computerization of other elements involved in the scheduling of vehicles and manpower. Most mail trucks are assigned to multiple services for a day's work, and it is critical that the timing allowed for each service is adequate and efficiently allocated. Relay bundle runs have to transport the bundled mails to the relay boxes in time for mail carriers to distribute according to their

schedules. Street letter box collections periods are restricted both by the delivery and the process plant requirements. The geographical boundaries of each street letter box collection and relay bundle run are, therefore, important to the structure of the vehicles schedules.

1.2 Purpose of the Study

With the advent of computerization, it is ideal to have a mechanized system that can solve all the complex scheduling problems in a short time. A complete and detailed computer scheduling program at the present stage is unfeasible because of the inability to comprehend all the details of an old existing manual system in a short period. ROUTPLOT, a computerized routing program used in the 1975 vehicle utilization study proves that although this program does not consider every minor details such as traffic restrictions, stop signs, etc. in the construction of the route, it has relieved the management from spending a significant amount of time in routing or re-routing of the vehicles schedules once alterations are induced. Modification of the computer results is necessary, and the resultant schedules can be carried out within a day's time instead of a week's or month's time to change the routings manually.

The routing of a vehicle actually depends highly on the geographic boundaries of the assigned schedule.

Traditionally, the boundaries are either natural geographic boundaries confined by rivers, highways, bridges and sea or are arbitrarily determined postal zones. The boundaries for areas with little physical hindrance are primarily set by experienced planners who are ex-truck drivers. This system of boundary determination is in fact very reliable, but time-consuming. The purpose of this study is two-fold:-

(1) to examine the characteristics of 12 clustering techniques; and

(2) to validate, using contrived and empirical data, the applicability of these 12 techniques in grouping box locations into suitable cluster.

1.3 Overview

Chapter I begins with general introduction supplemented by a brief description of the Vancouver City Postal Transportation Service operations. The purpose of the study is then outlined and an overview completes the first chapter.

Chapter II presents a broad view on cluster analysis by stressing the need of clustering algorithms and the conceptual problem in utilizing cluster analysis. Elements of the clustering analysis such as variables, scale and measures are discussed to indicate the variety of methods in amalgamating variables and constituting data set for cluster algorithms. A review of clustering techniques and

their uses, and some general remarks on the utilization of cluster analysis complete this chapter.

Chapter III begins with a brief introduction to the 9 hierarchical clustering methods examined in this study. The basic approach to these techniques is first described and the characteristics and criterion of each technique, with specific reference to distance measures are then discussed in details in the second part of this chapter. The rationale behind each method is also reviewed by referring to literature on these methods.

Similar to the above chapter, Chapter IV discloses the philosophy, characteristics and criterion of three non-hierarchical clustering methods that are applicable to distance measured data and variables.

In Chapter V, the results generated by twelve different clustering techniques on four sets of input data are examined and evaluated in detail. Two sets of contrived data are designed to test the applicability of techniques for evenly and unevenly distributed data sets. The other two data sets are actual street letter box locations of North and South Burnaby, and they are used to test the appropiateness and efficiency of these clustering techniques as aids to the scheduling problem. Statistical analyses on

the travel times and distances of each set of results are performed and used as evaluation measures.

The concluding chapter will discuss the suitability of the tested clustering techniques as a tool for scheduling street letter box collections and perhaps bundle relay runs for the Post Office. Areas of additional investigation are also included in this chapter.

CHAPTER II

CLUSTERING ANALYSIS : A BROAD VIEW

This chapter gives an introduction to the subject of cluster analysis. In general terms, cluster analysis can be referred to as a collection of techniques adopted in different applied fields to classify objects into groups which satisfy some criteria of homogeneity, inter-relatedness, or inter-group separation. This collection of techniques is the contribution of scientists from various fields, each holding a different viewpoint on the topic of classification or clustering. The variety of techniques, uses and measures of clustering analysis were developed to suit diverse purposes in grouping data, variables or objects into categories usable for further data interpretation or other mathematical analysis. There is no general solution to the problem of clustering, partly because the application of each criterion of performance leads in principle to a different outcome. It is, therefore, necessary to understand the concepts behind each clustering algorithm, to investigate the applicability of different approaches, and to choose the correct method to derive usable clusters for a set of objects.

2.1 Need for Clustering Algorithms

Classifying objects or data into groups with defined criteria or intuitive notions requires numerous enumerations to search all the possibilities and to choose the best solution. This enumeration process would be very time-consuming and difficult. Abramowitz and Stegun (1968) indicated that the number of ways of sorting n observations into m groups is a Stirling number of the second kind

$$S_n^{(m)} = \frac{1}{n!} \sum_{k=0}^{k=m} (-1)^{m-k} \binom{m}{k} k^n .$$

For even the relatively tiny problem of sorting 25 observations into 5 groups, the number of possibilities is the astounding quantity of over 2×10^{15}. This number could be further compounded if number of ideal groupings is not known prior to this enumeration.

In order to simplify the complexity in sorting objects into appropiate number of groupings, clustering algorithms are used. These algorithms are procedures for searching through the set of all possible clusters to find one that fit the data reasonably well. Frequently, there is a numerical measure of fit which the algorithm attempts to optimize, but many useful algorithms do not explicitly optimize a criterion. These algorithms use different mode of search by sorting, switching, joining, splitting, adding

and searching the data set to identify the cluster that suits the criterion best. The choice of algorithm, however, is basically determined by the user's selection of the data unit, the variables and the similarity measures.

2.2 Conceptual Problems in Cluster Analysis

Application of cluster algorithms would introduce a host of problems even though the intuitive idea of clustering is clear enough. The foremost difficulty is that cluster analysis is only a collection of heuristic procedures with a variety of decision rules and algorithms. Series of intuitive decisions are required to determine which elements of cluster analysis repertory should be utilized. Unfortunately, the literature on cluster analysis does not provide a general framework for this collection of techniques that shows the steps involved, available alternatives, decision points, and relevent criteria for selecting among options. In the following sub-sections, the author attempts to build a framework from which the elements of cluster analysis could be easily related.

2.2.1 The Objective Function

Though there is no absolute solution to cluster problems, it is usually used to determine a partitioning that satisfies some optimality criterion. This optimality

criterion may be given in terms of a functional relation that reflects the levels of desirability of the various partitions or groupings. This functional relation is often termed objective function. Each algorithm uses a particular criterion:- distance measures, similarity or dissimilarity measures, or quantifiable measures of homogeneity are all adopted as objective criteria for different techniques. This variety constitutes the diversified problem of choosing appropriate data units, variables as well as measures of functional relations.

2.2.2 Choice of Data Units and Variables

The actual mechanics of the cluster analysis are performed on a sample of entities representing "objects", "observations" or "elements". These entities could be the entire small single population or the fraction of a large population. If random samples were chosen from a large population to represent the population, independence of the data must be assumed. Cluster analysis on a given data set reflects the characteristics of these data units, thus the choice of data affects the outcome of the analysis.

Another problem facing the user is the choice of variables that can identify the entity's characteristics, attributes or traits. Any relevant discriminating variable could highly affect the result of the cluster analysis.

Missing variables could well generate amorphous and confusing clusters. On the other hand, inclusion of strong discriminators not relevant to the purpose at hand could mask the sought-for clusters and give misleading results. A selection method must be used to determine relevant variables, and based on this selection of variables and with proper scaling, a single index of similarity could be derived.

In most statistical theory discussions, the variables are always assumed to be of a single type, usually continuous and on an interval scale. This convenient assumption, of course, fully increases the power of mathematical techniques. However, in real world problems, variables are usually of mixed types. In general, variables can be classified according to the size of the range set or the scale of measurement, and they can be cross-classified[2]. In many cases, mathematical formulation and transformation are used to convert differently classified and scaled variables into usable forms for further interpretation and manipulations. This need of transformation induce the formulation of scale conversions.

The homogeneity of scale types, as mentioned in above paragraph, is required in most analysis techniques. Transformation of scale types, however, must be evaluated by its importance and suitability to the analysis techniques.

Scales are usually referred to as nominal, ordinal, interval or ratio measures. The transformation from one type of scale to another often involves subjective consideration of the validity of conversions. For cluster analysis, the variables are usually quantifiable and unquantifiable scales are transformed by various methods (Cochran and Hopkins, 1961; Shepard, 1962a, 1962b; Kruskal, 1964; Anderberg, 1973) to suit the requirements of clustering techniques.

2.2.3 Measures

The majority of clustering techniques begins with the calculation of a matrix of similarities or distances between entities, and therefore consideration is needed of the possible ways of defining these quantities. Indeed many clustering techniques may be thought of as attempts to summarize the information or relationships between entities which are given in a similarity matrix, so that these relationships can be easily comprehended and communicated.

Although variables could be scaled by various transformations, a measure of association is still needed to relate, in numerical form, the similarity of one variable to another. This measure is required because all clustering methods have the same basic working assumption that numerical measure among data or variables are comparable and sortable. Different types of variables warrant various measures of

associations. There is a host of methods for calculating the measures among variables:- the angular measure between vectors, the product moment correlation coefficient, the canonical correlation, the matching coefficients, and some probability-based measures are a few measures commonly adopted in establishing numerical relations among variables. These measures of association of variables, however, need to be supplemented by a measure of association of data units if the clustering technique so chosen is designed for grouping data units.

The measures of association among data units differ from that of variables in many aspects. The most prominent difference is that the measure for data unit warrants a sometimes nonexisting relationship measure among the variables related to the data unit. The heterogeneity of variety of measurement units and variable types makes it especially difficult to define meaningful measures of association between data units, within the context of a given set of variables. Similarity and distance measures are the most popular measures adopted in clustering data units. There are a number of similarity measures, as well as distance measures that are applicable to binary and qualitative data (Anderberg, 1973; Everritt, 1974; Duran, 1974). Two dimensional problems are easier to measure: the distances among data units could be simply Euclidean distances. Multi-

dimensional problems, however, require experimentation to examine the validity of weight assignment, representation spaces, and the basic approach to formulate such a measure.

2.2.4 Other Problems of Cluster Analysis

Even though the user has decided the objective function, data and variables to be used, and the similarity measure between data or variables, there are still three big questions to be considered:- 1. what to cluster; 2. number of clusters; and 3. the choice of technique.

The variables are often amalgamated into a single index related to the data unit and it is sometimes necessary to categorize the objects by variables instead of data units. Similar to factor analysis, the multivariate data are often classified and grouped by the attributes of different variables separately or simultaneously. This choice could depend on the degree of difficulty in amalgamating the variables and the relevance of individual variable to the cluster analysis.

A substantial practical problem in performing a cluster analysis is deciding upon the number of clusters in the data. Different clustering methods offer various degrees of flexibility in grouping numbers of clusters. Hierarchical clustering methods give a configuration for every member of

the cluster from one up to the number of entities whereas other approaches might require defined number of clusters prior to the clustering procedures. Some algorithms begin with a chosen number of groups and then modify this number as indicated by certain criteria with the objective of simultaneously determining both the number of clusters and their configuration. All these indicate that the choice of technique could affect all the elements of cluster analysis.

The choice of technique is an inherent problem in using cluster analysis. As mentioned previously, the collection of clustering techniques were developed by scientists of different fields to satisfy their own needs; and each technique would be particularly suitable to one set of data or criteria. A review of literature would help to determine the technique required for sorting objects with appropriate criteria and algorithms. Further discussions of clustering techniques is included in section 2.3 of this chapter.

2.3 A Review of Clustering Techniques

The various background of scientists and researchers who developed different clustering techniques results in a variety of clustering algorithms. Comprehensive reviews of classification methods was conducted by Cormack (1971) and Anderberg (1973). In general, cluster analysis techniques can be "classified" into types roughly as follows[3]:-

(i) Hierarchical techniques - in which the classes themselves are clustered into groups, the process being repeated at different levels, step by step, to form a tree diagram.

(ii) Optimization - partitioning techniques -- in which the clusters are formed by optimizing the clustering criterion. The classes are mutually exclusive, thus forming a partition of the set of entities.

(iii) Density or mode-seeking techniques -- in which clusters are formed by searching for regions containing a relatively dense concentration of entities.

(iv) Clumping techniques -- in which the classes or clumps can overlap.

(v) Others -- methods which do not fall clearly into any of the four previous groups.

Of all the above categories, hierarchical techniques are most commonly used and discussed. This category of techniques may be subdivided into "agglomerative" methods which proceed by a series of successive fusion of the N entities into groups, and "divisive" methods which partition the set of the N entities successively into finer partitions. The results of both types can be presented in the form of a dendogram or a two dimensional tree diagram, illustrating the fusions or partitions which have been made at each successive level. Both types of hierarchical techniques can

be viewed as attempts to find the most efficient step, in some defined sense, at each stage in the progressive subdivision or synthesis of the population. Further discussion of hierarchical techniques is included in Chapter III.

Besides hierarchical clustering techniques, the others could be simply termed as non-hierarchical methods. Partitioning techniques can be formulated as attempts to partition the set of entities so as to optimize some predefined criterion. Most of these methods assume that the number of groups had been decided by the user, although some allow the number to be changed during the course of analysis. Three distinct procedures are employed by these techniques:

(a) a method of initiating clusters;

(b) a method for allocating entities to initiated clusters; and

(c) a method of reallocating some or all of the entities to other clusters once the initial classificatory process has been completed.

These non-hierarchical techniques are further examined in Chapter IV.

Density search techniques originated from single linkage cluster analysis. These techniques locate the high density regions and define them as clusters (Carmichael, 1968).

Clumping techniques allow overlaps between classes indicating that the overlapped entities may belong in several places (Jones and Jackson, 1967). Other methods such as the Q-factor analysis (Cattell, 1952; Parks, 1970; Johnson, 1970), R-factor analysis (Gower, 1966), BC Try System (Tryon and Bailey, 1970), and many others are less known among clustering techniques. Most clustering methods warrant extensive enumeration, and computer programs are written to perform these procedures efficiently with high accuracy.

2.4 Uses of Clustering Techniques

Cluster analysis is generally used to sort data, variables or objects into meaningful classes for further interpretation. Since the characteristics and criterion of each clustering technique are designed differently for various purposes, it is hard to define the limits of cluster analysis. On one end of the spectrum, it resembles the factor analysis procedures, and on the other, it is nothing but a sortation methodology. Applications of clustering techniques, therefore, vary from simple sortation of one- or two-dimensional data sets to complex multi-dimensional data classifications. The cluster analysis is most widely used by biological scientists in classifying species of different families. Other fields, such as psychology, geological sciences, economics, archeology, medicine and many other use cluster analysis mainly in sortation of multivariate data sets.

The development of computer technology has helped to reduce the computational time for allocating or reallocating clusters within a data set. The increasing interest in applied statistics and the availability of vast amount of data have further emphasized the importance of selecting, sorting and grouping useful data into clusters for other mathematical analysis. The uses of cluster analysis, however, is often tempered by the difficulties in interpretating the results. These difficulties could be the results of:-

(i) the failure to recognize the inappropiateness of techniques on the set of data by the user; and

(ii) the ignorance of the possibility of the absence of clusters in the data set.

In using cluster analysis, it is necessary to keep in mind that:

(a) clustering techniques do not optimize with respect to the criteria;

(b) selection of variables, data, and measures is critical to results; and

(c) bias opinion on data and variables selection of a sampled population could generate confusing and meaningless results.

CHAPTER III

HIERARCHICAL CLUSTERING TECHNIQUES

The brief review of clustering techniques in Chapter II has indicated that there is a host of clustering methods applicable to data, variables or objects classifications. Hierarchical clustering techniques are the most commonly used and, by far, the most discussed in literature. It is difficult to examine every variation of hierarchical clustering techniques disclosed in the literature, and in this study, only 9 of the more popular hierarchical methods are reviewed. The characteristics, criterion and problems, if any, associated with each of these techniques, with specific references to agglomerative approach, distance measure, and data clustering, are examined in the following sections.

The abundance of hierarchical clustering methods treated in the literature are alternative formulation or minor variations of three basic clustering concepts[4]:-

(1) Linkage methods,

(2) Centroid methods, and

(3) Error sum of squares or variance methods.

All these methods are compatible for clustering data units and only linkage methods can be adopted to variable clustering procedures.

3.1 Basic Agglomerative Procedure and Approaches

The basic procedure with all the agglomerative methods is similar. They generally start with the computation of a correlation or distance matrix between the entities, and the end product is a dendogram showing the sequential fusions of individual entities.

The correlation or distance matrix, which contains the association measure, S_{ij} or d_{ij}, between entities i and j, is the most important component of the clustering procedure. Most procedures assume the measure between entities is symmetric i.e. $S_{ij} = S_{ji}$ or $d_{ij} = d_{ji}$, and because of this assumption, only the lower triangle of the correlation or distance measure is utilized in the clustering procedures. One critical point concerning the elements of the matrix is that the clustering procedures are not designed to handle negatively valued elements, thus, the absolute values or square of the measure are frequently used as the association measure. Another aspect related to the clustering procedure is that there are significant differences in the characteristic of a correlation and a distance matrix of n entities:-

(a) Pairwise distances $d(x_i, x_j)$ may be represented in terms of a symmetric $n \times n$ distance matrix:-

$$D = \begin{pmatrix} 0 & d_{12} & \cdots & d_{1n} \\ d_{21} & 0 & \cdots & d_{2n} \\ \vdots & \vdots & & \vdots \\ d_{n1} & d_{n2} & \cdots & 0 \end{pmatrix}.$$

The diagonal elements of the matrix D are $d_{ii} = 0$ for $i = 1,2\ldots,n$ and $d_{ij} > 0$ for $i,j = 1,2\cdots,n$.

(b) The correlation measure S_{ij} between two entities is non-negative real valued function subjected to[5]:-

(i) $0 \leq S_{ij} < 1$ for $i \neq j$;

(ii) $S_{ii} = 1$ for $i = 1,2,\cdots,n$; and

(iii) $S_{ij} = S_{ji}$.

The pairwise correlation can be represented in a matrix of

$$S = \begin{pmatrix} 1 & S_{12} & \cdots & S_{1n} \\ S_{21} & 1 & \cdots & S_{2n} \\ \vdots & \vdots & & \vdots \\ S_{n1} & S_{n2} & \cdots & 1 \end{pmatrix}$$

These marked differences of distance and correlation measure would reverse the clustering criteria of some hierarchical methods described in the following sections.

Once the matrix is defined, the basic agglomerative approach in clustering entities into groups can be considered as follows:-

(1) Begin with n clusters each consisting of exactly one entity. Let the clusters be labeled with the numbers 1 through n .

(2) Search the correlation or distance matrix for the pair of clusters that satisfies best the clustering criterion. Let the chosen clusters be labeled p and q and let the association measure be S_{pq} or d_{pq}, $p>q$.

(3) Reduce the number of clusters by 1 through merge of clusters p and q . Label the product of the merge q and update the correlation or distance matrix entries in order to reflect the revised association measures between cluster q and all other existing clusters. Delete the row and column of the original matrix pertaining to cluster p .

(4) Repeat steps (2) and (3) for (n -1) times to form one single cluster containing n entities. The identity of the clusters that are merged and the values of measures between them at each stage are recorded for the final dendogram output.

Various agglomerative methods differ from this basic procedure in using the clustering criterion to define the most closely associated pair at step 2 and in updating and revising the correlation or distance matrix at step 3. These variations would produce drastically different results for some data set and little or no variance for others.

This basic agglomerative procedure is actually a series of comparisons between entities and based on these comparisons, clusters are formed. Comparison of matrix elements, searching of pair of clusters, update the similarity matrix, and deletion of matrix rows all added up to a total of $2n^2 - 9n/2$ comparisons[6]. The number of comparisons should be one of the considerations for the size of the input matrix and the timing required for computation.

There are several computational approaches to clustering problems. Each approach has its own unique advantages and limitations. No single approach is a cure-all for all circumstances; each has its own realm of applications. Among the computational approaches, "stored matrix", "stored data" and "sorted matrix" are more commonly used.

Stored matrix approach involved the storing of the correlation or distance matrix in the computer's central memory so that the similarity values may be accessed directly in any sequence. This approach, like any others has its own unique characteristics:-

(1) The clustering procedure is independent of the derivation of correlation or distance matrix;

(2) This in-core storage method severely limits the number of entities that can be grouped. Anderberg (1973) indicates that problems of more than 150 entities are difficult

to handle without a large size computer. The storage requirements for correlation or distance matrix are shown in Table 1.

Number of entities	Storage required	Number of entities	Storage required
50	1,225	300	44,850
100	4,950	350	61,075
150	11,175	400	79,800
200	19,900	450	101,025
250	31,125	500	124,750

Table 1. Storage Requirements for Similarity Matrix[7]

(3) Both variables and data to be grouped are represented in the matrix and do not affect the procedure of clustering algorithm in this stored matrix approach.

In using this approach, the user must be aware of the capacity of the computer for storage and enumeration process of different methods.

Stored data approach involved more computational procedures in its algorithm than the stored matrix approach. The storage of data, instead of correlation or distance matrix elements, in the central memory requires less core space.

This approach is applicable to any clustering methods with similarity or distance measures, and the combinatorial problem in computing association measures between clusters by reference to the original data is comparable to the sizing issue of stored matrix approach. This computational problem can be avoided by storing either the association values or the summary statistics for each cluster from which desired association measure could be computed in the computer memory. This remedy, however, has restricted the user of this approach to data unit clustering.

Sorted matrix approach is a relatively unexploited methodology. This is designed to handle sizeable similarity matrix for sorting data units or variables. The distinct advantage of this approach is that it saves computer storage space but the matrix has to be sorted before being input to the clustering program.

Other approaches are also available by using various combination of the above approaches or specially designed algorithm for specific computer systems. Magnetic tapes and disks are mostly used by other approaches (Wishart, 1969b; Park, 1970; Wolfe, 1970) for storing data units and/or similarity matrix, thus up to 1000 data units and 200 variables can be clustered with hierarchical methods. Variations of stored matrix and stored data approaches are generally adopted in most computational methods handling large data set.

3.2 Linkage Methods

This category of hierarchical clustering methods is simple to use and easy to understand. This classification procedure is essentially identical to that of basic agglomerative procedure. Maximum, minimum or average values of the association measures among entities are used as criteria for grouping the data units or variables into clusters. Methods using different clustering criterion, of course, result differently in the dendograms. In the following subsections, the characteristics of four linkage methods are described.

3.2.1 Single Linkage Methods

The methods of single-linkage cluster analysis are the simplest of all hierarchical techniques, and are also the most popular. This approach was first described by Sneath (1957) and later by many other scientists (McQuitty, 1960; Lance and Williams, 1966; Johnson, 1967; Gower and Ross, 1969; Zahn, 1971; Sibson, 1973; Hartigan, 1975). The criterion used in these techniques is the minimum distance (maximum value if correlation measure is used) between clusters. At each stage, after clusters p and q have been merged, the similarity between the new cluster t and some other cluster r is determined by

$$d_{tr} = \min (d_{pr}, d_{qr}) \text{ , or}$$

$$s_{tr} = \max (s_{pr}, s_{qr})$$

The measure between the two closest or most similar members of clusters t and r is the criterion for further merge. This criterion is used throughout the enumerations until one single cluster is formed.

This linkage procedure is known as single linkage because clusters are joined at each stage by the single shortest or strongest link. For any cluster of two or more entities produced by this method, every member is more similar to some other member of the same cluster than to any other entity not in the cluster. However, this approach is often criticized for its resultant chaining clusters for non-ellipsoidal groupings. It is frequently stated that this "chaining" has distinctly dissimilar entities at each end of the cluster. The uses of different distance measures for this approach, however, would yield different "chaining" effects.

Distance measures such as simple "City-Block" and Euclidean distances, and Chi-sqares are often used in this approach to cluster data sets. The outcomes in using these different measures, of course, are different and are classified as three different single linkage methods in this study.

3.2.2 Complete Linkage Method

Different from single-linkage method, the complete linkage method uses maximum distance or least correlation as criterion to group entities. Sorensen's (Sneath, 1968) complete linkage criterion is that two individuals in a group have a similarity or distance which is less than a threshold value s or r. Other scientists termed this method as furthest neighbour technique, in which each individual is treated as a single-point cluster. This approach is considered as "masimally connected subgraph" in graph theory.

Similar to single-linkage procedure, at each stage of complete linkage algorithm, after clusters p and q have been merged, the association measure between the new cluster t and some other cluster r is determined as $d_{tr} = \max(d_{pr}, d_{qr})$ for distance measure, and $S_{tr} = \min(S_{pr}, S_{qr})$ for correlation measure. The quantity d_{tr} (or S_{tr}) is the distance (or correlation) between the most distance (or dissimilar) members of clusters t and r. If clusters were merged, then every entity in the resulting cluster would be no further than d_{tr} or more than S_{tr} from every entity in the cluster. The d_{tr} or S_{tr} can be considered as the diameter of sphere of which the maximum distance or minimum correlation is related to.

The interpretation of the clusters, in contrast to single-linkage method, can be made only in terms of the relationship within individual clusters; and there is no particularly useful interpretation involving the differences between clusters.

3.2.3 Average Linkage Within the New Group

Instead of relying on extreme values, maximums or minimums, used in single-linkage and complete linkage methods as criteria for grouping entities into cluster, average linkage method utilizes average values of the measures as a rule for grouping entities or clusters. Two methods employ this criterion: one uses the within group averages and the other compares the between merged group averages. The latter method is discussed in section 3.2.4.

The d_{ij} or s_{ij} entries in the initial similarity matrix may be built as the sum of similarities associated with all pairwise combinations formed by taking one entity from cluster i and the other from cluster j. Prior merges of any entities, each cluster consists of just one single entity and there is only one such pair of entities for each pair of clusters. Upon the merges of clusters p and q, the sum of pairwise similarities between the new cluster t and some other cluster r becomes:

$$d_{tr} = d_{pr} + d_{qr} \quad \text{for distance measures}$$

or $$s_{tr} = s_{pr} + s_{qr} \quad \text{for correlation measures}$$

and the similarity matrix is updated accordingly.

The sum of all pairwise similarities among entities within cluster i, SUM_i becomes:

$$SUM_t = SUM_p + SUM_q + d_{pq}$$

when cluster p and q are merged and new cluster t is formed. At the same time, the number of entities N_i for cluster i increases accordingly as:

$$N_t = N_p + N_q$$

In searching for the most similar pair, the average within group similarity for the clusters formed by merging the candidate pair of cluster i and j becomes

$$\frac{SUM_i + SUM_j + d_{ij}}{(N_i + N_j)(N_i + N_j - 1)/2} .$$

This average linkage method has not made any reference to the maximum or minimum similarity values and the interpretation of the resulting dendogram would need a different approach than that for the single or complete linkage results. However, as a practical matter, this method frequently gives results that are little radical different from those obtained with complete linkage method[8].

3.2.4 Average Linkage Between Merged Groups

This average linkage method uses different average values from that of the above method. Similar to the average linkage within group technique, this method defines distance between groups as the average of the distance between all pairs of individuals in the two groups. The procedure can be used with correlation and distance measures as long as the concept of an average measure is acceptable. The similarity matrix contains d_{ij} (or S_{ij}), the sum of similarities associated with all pairwise combinations between cluster i and j. The number of such between group pairwise similarities is the product of N_i and N_j where N_i is the number of entities in cluster i. The average between group similarity for cluster i and j can be formulated as

$$\frac{d_{ij}}{N_i\ N_j} \quad \text{or} \quad \frac{S_{ij}}{N_i\ N_j} \ .$$

In this method, the sums of within group pairwise similarities are ignored. In reference to application to this method to correlation measure matrix clustering, Lance and Williams (1967) point out that using $\cos\left[\frac{1}{N_i N_j} \sum_{ij} \cos^{-1} S_{ij}\right]$ as a similarity measure would be more appropiate.

3.3 Centroid Methods

These methods merge clusters with the most similar mean vectors or centroids. Two different approaches were developed by scientists: centroid clustering analysis (Sokal and Michener, 1958; King, 1966 and 67; Lance & Williams, 1967a) and median method (Gower, 1967; Lance & Williams, 1967a). The first method employs weighted measures according to the number of entities in the formulation of mean vectors whereas the latter method uses equal weighs for centroids of groups. An unique characteristic of the centroid methods and their variants is that the similarity value associated with the mergers of the most similar cluster may rise and fall from stage to stage. This is the reversal phenomenon associated with this approach. These reversals occur because cluster centroids can migrate as mergers take place.

3.3.1 Centroid Method

This method was originally proposed by Sokal and Mitchener (1958) and King (1966, 1967) who concentrate on the clustering of variables. Groups are depicted to lie in Euclidean space, and are replaced on formation by the coordinates of their centroid. The distance between groups is defined as distance between the group centroids. The procedure is then to fuse groups according to the distance between their centroids, the groups with the shortest distance being fused first.

Lance and Williams (1967a) update the formulation of distance measure between centroids as:-

$$d_{tr} = \frac{N_p}{N_p + N_q} d_{pr} + \frac{N_q}{N_p + N_q} d_{qr} - \frac{N_p N_q}{N_p + N_q} d_{pq}$$

where p and q are the labels for the clusters just merged, t is the label for the new cluster, and r is any other existing clusters. This equation could be used with any similarity measure for either variables or data units, however, the results would lack a useful interpretation if d_{ij} is not the squared Euclidean distance between the centroids of cluster i and j .

3.3.2 Median (Gower) Method

A disadvantage of the Centroid Method is that if the sizes of the two groups to be fused are very different, the centroid of the new group will be very close to that of the larger group and may remain within that group; the characteristics of the smaller group are then virtually lost. The strategy can be made independent of group size by assuming that the groups to be fused are of equal size, the apparent position of the new group will then always be between the two groups to be fused. In other words, as proposed by Gower (1967) the general idea is that the centroids are weighted equally regardless of how many entities are in the respective clusters. When d_{ij} is a distance function, the updating

equation for the Median method is

$$d_{tr} = \tfrac{1}{2} (d_{pr} + d_{qr}) - \tfrac{1}{4} d_{pq}$$

or $$S_{tr} = \tfrac{1}{2} (S_{pr} + S_{qr}) - \tfrac{1}{4} (1 - S_{pq})$$

if S_{ij} is a correlation function.

Although this method could be made suitable for both similarity and distance measures, Lance & Williams (1967a) suggest that it should be regarded as incompatible for correlation measure, since geometrical representation of the measure cannot be interpreted easily.

3.4 Error Sum of Squares or Variant Methods

Although there are several methods using error sum of squares as objective function for clustering entities, they are variations from the method developed by Ward (1963) and Ward and Hook (1963). In this study, only Ward's method is examined.

Ward (1963) proposes that at any stage of an analysis the loss of information which results from the grouping of individuals into clusters can be measured by the total sum of squared deviation of every point from the mean of the cluster to which it belongs. At each step in the analysis, union of every possible pair of clusters is considered and the two clusters whose fusions results in the minimum increase

in the error sum of squares are combined. In the formulation of this approach, the following are defined and calculated:-

X_{ijk} = score on i^{th} of n variables for j^{th} of m_k data units in k^{th} of h clusters

$$\bar{X}_{ik} = \sum_{j=1}^{j=m_k} X_{ijk}/m_k$$

= mean on the i^{th} variable for data units in k^{th} cluster

$$T_{ik} = \sum_{j=1}^{j=m_k} X_{ijk} = m_k \cdot \bar{X}_{ik}$$

= total of scores on i^{th} variable for data units in the k^{th} cluster

$$S_k = \sum_{i=1}^{i=n} \sum_{j=1}^{j=m_k} X_{ijk}^2$$

= sum of squared scores on all variables for all data unit in the k^{th} cluster

Then the error sum of squares for cluster k may be written as

$$E_k = S_k - \sum_{i=1}^{i=n} T_{ik}^2/m_k$$

The increase in the total error sum of squares due to the merger of clusters p and q to form the new cluster t is

$$\Delta E_{pq} = E_t - E_p - E_q$$

$$= S_p + S_q - \sum_{i=1}^{i=n} (T_{ip} + T_{iq})^2 / (m_p + m_q) - E_p - E_q$$

Based on the above formulas, entity with least ΔE_{pq} is grouped into the new cluster.

Wishart (1969a) in his computer algorithm indicates that the variables with a large variance have more influence on the joins than those with a small variance. The joining of units and sub-groups is decided on the basis of the contributions to the sum by the squares deviation E_k. Although the joining of sub-groups p and q may result in a smaller sum of squared deviations, a link between p and r is decided because the increase of error sum of squares ΔE_{pr} is less than that of ΔE_{pq}. This occurrence disrupts the homogeneity of the groups entities. Alteration to both the sub-group structure and the changes in error sum of squares are required in formulating this algorithm.

The Ward method is designed for a similarity matrix of Euclidean distances computed in any decided representation space. Although this method may or may not give the minimum possible error sum of squares over all possible sets of h clusters from the m data units, the solution is usually very good even if it is not optimal on the criterion.

3.5 Summary

Many of the above mentioned hierarchical clustering methods, using distance measure between groups as criterion, can be represented as a recurrence formula for the distance between a group k , and a group (ij) formed by the fusion of groups i and j . This formula can be written as:-

$$d_{k(ij)} = \alpha_i d_{ki} - \alpha_j d_{kj} - \beta \cdot d_{ij} - \gamma |d_{ki} - d_{kj}|$$

When d_{ij} is the distance between groups i and j and α, β and α are parameters related to different methods as shown in Table 2.

Single Linkage: $\alpha_i = \alpha_j = 1/2$; $\beta = 0$; $\gamma = -1/2$

Complete Linkage: $\alpha_i = \alpha_j = 1/2$; $\beta = 0$; $\gamma = 1/2$

Centroid: $\alpha_i = n_i/(n_i+n_j)$; $\alpha_j = n_j/(n_i+n_j)$; $\beta = -\alpha_i \alpha_j$; $\gamma = 0$

Median: $\alpha_i = \alpha_j = 1/2$; $\beta = 1/4$; $\gamma = 0$

Ward's Method:

$$\alpha_i = \frac{n_k + n_i}{n_k + n_i + n_j}\ ;\ \alpha_j = \frac{n_k - n_j}{n_k - n_i - n_j}\ ;$$

$$\beta = \frac{-n}{n + n_i + n_j}\ ;\ \gamma = 0$$

Table 2. Parameter Values of the Recurrence Formula for Five Hierarchical Techniques

This recurrence relationship is given by Lance & Williams (1967a) and by Wishart (1969c) and it is not suitable for methods using correlation measures.

On the whole, hierarchical clustering techniques all have their merits and limitations. There is no single method that would solve all types of clustering problems and it is necessary for the user to examine the suitability of these techniques for grouping the data sets.

CHAPTER IV

NONHIERARCHICAL CLUSTERING TECHNIQUES

For a data set of n entities the hierarchical methods give n nested classifications ranging from n clusters of one member each to one cluster of n members. Contrary to this, nonhierarchical techniques introduced in this chapter are designed to cluster data units into single classification of k clusters, where k is either specified prior to the procedures or determined as part of the clustering method. These methods may be used with much larger problems than the hierarchical methods because it is not necessary to calculate and store the similarity or distance matrix; it is not even necessary to store the data set. In general, the data units are processed serially and can be read from tape or disk as needed; and this characteristic allows clustering of larger collections of data units.

In this study, only three of the nearest centroid sorting methods with fixed number of clusters are examined in details.

4.1 Elements of Nonhierarchical Methods

Most nonhierarchical procedures can start with initial partitions or initial seed points of the data units. In using initial partitioning, the algorithms change the cluster memberships into "better" partitions. The broad concept for these methods is very similar to that underlying the steepest descent algorithms used for unconstrained optimization in nonlinear programming[9]. These methods start with initial seed points and then generate a sequence of moves from one point to another, each giving an improved value of objective function, until a local optimum is found. The seed point and initial partition are, therefore, important to the nonhierarchical methods. These initial configurations can be chosen randomly or methodically as discussed in the following sub-sections.

4.1.1 Seed Points

Various approaches are used in choosing a set of seed points that are adopted as cluster nuclei around which the set of n data units can grouped. Some methods use data units themselves as seed points, whereas other use more sophisticated methodology in arriving the nuclei.

The simplier methods choose (1) k data units from the set randomly (McRae, 1971); (2) the first k data units in the data set (McQueen, 1967); (3) the labeled $n/k, 2n/k, \cdots, (k-1)n/k$ and n data units which are initially tacked on as 1^{st} to n^{th} data units; or (4) subjectively k units from the data set. More calculated approaches for selecting the seed points use centroids of initial partitions (Forgy, 1965), densities of initial groups (Astrahan, 1970) or mean vectors of the data set (Ball and Hall, 1967). These approaches, like any other elements of clustering techniques, have a substantialy different influence on the results of clustering procedure.

4.1.2 Initial Partitions

In lieu of seed points, some clustering methods emphasize on the generating of intial partition of the data units into mutually exclusive clusters. However, the set of initial seed points are required to generate initial partition in some of the partitioning procedures.

Forgy (1965) uses a given set of seed points as the nuclei to initiate a partitions formed by assigning the data units to the nearest seed point. The seed points remain stationary throughout the assignment of the full data set and consequently the resulting set of clusters is independent of the sequence in which data units are assigned. These clusters are separated by pairwise linear boundaries which are equi-

distant from the clusters nuclei in two dimensional problems.

MacQueen (1967) assigns data units one at a time to the initially single point clusters pre-defined by the seed points with the nearest centroid; centroids as the true mean vectors of all the data units are updated as the clusters' sizes grow. In this method the cluster centroids migrate so the distance between a given data unit and the centroid of a particular cluster may alter widely during the assignment process, as a result, the set of initial clusters is dependent on the order in which data units are assigned.

Wolfe (1970) uses Ward's hierarchical clustering method to provide an initial set of clusters for his algorithm. This approach, however, involve immense computational effort in setting up the partitions, thus limiting the size of the problem tremendously. Similar to Wolfe's approach, Lance and Williams (1967b) suggest using hierarchical methods on one or more subsets of convenient size and then use the resulting groups as nuclei for assignment of the remaining clusters.

Random assignment of partition, of course, is the simplest one to use. However, this approach would cluster entities without considering their homogeneity and thus is not an attractive alternative.

4.2 Nearest Centroids Sorting With Fixed Number of Clusters

Of all the nonhierarchical clustering techniques, the simplest iterative methods merely consist of two basic processes:- (1) a set of seed points are computed as the centroids of a set of clusters, and (2) a set of clusters can be constructed by assigning each data unit to the cluster with the nearest seed point. These two processes are repeated alternately until a stable configuration converges: a critical condition for completing clustering algorithms.

4.2.1 Convergence Properties

In using nearest centroid sorting algorithms, convergence of processes is critical and expected in grouping data units. Proofs for convergence are mostly rigorous and generally difficult to understand. On the whole, the total within group error sum of squares is the key to the convergence. Referring to the notation used in section 3.4, the total within group error sum of square E can be formulated as:-

$$E = \sum_{k=1}^{k=h} \left(\sum_{j=1}^{j=k} \sum_{i=1}^{i=n} (X_{ijk} - \bar{x}_{ik})^2 \right)$$

where $\sum_{i=1}^{i=n} (X_{ijk} - \bar{x}_{ik})^2$ is the squared Euclidean distance between the centroid of cluster k and the j^{th} data unit in that cluster.

Another characteristic in the algorithms that ensure convergence is the number of different ways a data set of n data units may be clustered into h clusters is a finite number if n is finite. This indicates that any method that generates each partition at most once is finitely convergent because there are only finitely many different partitions.

The criterion chosen for deciding convergence of the nearest centroid sorting method is the stability of cluster membership; an alternative criterion is stability of the cluster seed points. In most methods, the seed points are the cluster centroids which are dependent only on the cluster membership.

4.2.2 Forgy's Method

The simple algorithm suggested by Forgy (1965) consists of basically three steps:-

(1) Start with the desired initial configuration. If the configuration is a set of seed points, go to step 2; otherwise go to step 3.

(2) Assign data units to the clusters with nearest seed point. The seed points remain intact for a full cycle through the entire data set.

(3) Compute new seed points as the centroids of the cluster data units.

Steps 2 and 3 are repeated alternately until the process converges; that is, iterate until no data units change their cluster membership at step 2.

In view of the various initial configurations:- seed points or partitions, it is difficult to estimate how many iteration of the steps are required to achieve convergence in any particular problem. Empirical evidence reveals that five repetitions or less will be sufficient for small problems. A total of n distance computations and $n(k-1)$ comparisons of distances are required in assigning n data units to k clusters at each repetition of the two steps. Relatively limited number of iterations is actually necessary if the number of clusters is much smaller than the number of data units. This approach allows users to try several variations of the number of clusters at a less computational cost than for a full hierarchical analysis.

4.2.3 Jancey's Variant

Jancey (1966) suggests a method similar to Forgy's with a modified step 3. The first set of cluster seed points is either given or computed as the centroids of clusters in the initial position; at all suceeding stages each new seed point is formed by reflecting the old seed point through the new centroid for the cluster. This technique presumbly will accelerate convergence and possibly lead to a better overall

solution through bypassing inferior local minima. This approach, similar to Forgy's, implicitly minimizes the within group error function. The boundaries of the clusters are equidistant from each centroid and the result of this method is not affected by the sequence of data units within the data set.

4.2.4 Convergent K-Mean Method

Unlike Forgy's or Jancey's approach in assigning member to old centroids, MacQueen (1967) uses a "K-mean" process in allocating each data unit to the cluster with the nearest centroid computed on the basis of the cluster's current membership. The convergent clustering method (Wishart, 1969b; McRae, 1971) using the MacQueen's K-mean process can be implemented through the following sequence of steps[10].

(1) Begin with an initial partition of the data units into clusters. The partition could be constructed using any of the approaches described in section 4.1.2.

(2) Take each data unit in sequence and compute the distances to all cluster centroids; if the nearest centroid is not that of the data unit's parent cluster, then reassign the data unit and update the centroids of the losing and gaining clusters.

Repeat step 2 until convergence is achieved; that is until the membership in each cluster is stablized.

4.3 Summary

There are still several nonhierarchical methods described in the literature (MacQueen, 1967; Ball and Hall, 1965). Most other methods use similar procedures as the three described, but have different grouping criteria and updating procedures of the cluster elements. There are, undoubtly, limitations and merits in each of the other methods, and since these methods were developed for different fields, results from these techniques on the same set of data are expected to be different.

The three techniques mentioned in the previous sections all have the three distinct procedures of initiating, allocating and reallocating until convergence occur within the data set. In using these methods in clustering a given data set, the groupings are expected to be relatively similar. Forgy's and Convergent K-mean methods will probably give similar results because of their resemblance in procedures. Jancey's method, because of its different procedure in allocating new seed points, could produce unmatched clusters for the same given data set. The suitability of method application for a given data set, therefore, cannot be determined without trying all three methods, perhaps with different seed points or initial partitions and various number of clusters. Interpretation and subjective opinion on the results would be the only assets in evaluating the comparibility of these clustering methods.

CHAPTER V

COMPARATIVE EVALUATION OF CLUSTERING TECHNIQUES

Cluster analysis has always been an exploratory tool for generating hypothesis about the data or discerning fundamental facts previously not apparent. Interpretation of the results using various tools is needed to justify hypothesis or simply to evaluate the suitability of the techniques. This stage of judgement is subjective, intuitive, and heuristic. Comparisons of results generated by different techniques would indicate not only the relationship between entities but also the validity of the desired results. This chapter describes the evaluation process used in this study by examining the appropiateness of each of the twelve clustering methods to group four sets of two-dimensional data, and discusses the results so obtained.

5.1 Approach to the Evaluation Process

The above chapters have stressed that each element of a cluster analysis has its own importance to the actual clustering procedure. Pertinent information on the data set is by far the critical to the grouping procedures:- the number of variables, the variable scales, and the association of these variables to each other of a data unit are the key inputs to a clustering algorithm. The association measure between

data units is another essential element of the clustering analysis. The number of clusters, the cluster key and the clustering technique to be used are the other vital elements of a grouping analysis.

The scheduling problem of the Post Office has two unique characteristics:- the location of the boxes are fixed and the travelling speed of the trucks are standardized to be 15 m.p.h. These two conditions emphasize the need for an efficient technique to assign appropiate number of call points to each truck. The objective of this study is to find, if possible, such a technique. The comparative evaluation process is, therefore, constructed to investigate the appropiateness of several clustering techniques on data sets, measures and other elements that are pertinent to this scheduling problem.

5.1.1 Data Set

In order to examine the applicability of different clustering methods to data sets of various spatial characteristics, four sets of data pertinent to the Post Office scheduling problem are used. All the data units of these four sets have only two variables representing the x and y coordinate of a two dimensional Cartesian space. The use of this dimensional space is based on the assumption that the distances between data units are computable by using two variables. Both variables, in this case, are of interval type, and there is no need

to use scale conversions for generating conformity of variables. These four data sets can be identified as:-

(1) evenly distributed contrived data;

(2) unevenly distributed contrived data;

(3) empirical data for North Burnaby area; and

(4) empirical data for South Burnaby area.

The first set of data was constructed by allocating randomly 80 data points each representing the arbitrarily locality of a mail box. These data points are fairly evenly distributed, and there is no outstanding grouping which can be spotted visually (Figure 1).

The second set of contrived data is essentially a collection of data points falling into three visually identifiable groups (Figure 2). This set also contains 80 data points, and is designed to test the ability of each technique in outlining the visually feasible boundaries of the three groups.

Both the North Burnaby and the South Burnaby data sets represent the localities of mail boxes in the Municipality of Burnaby. The boundary dividing North and South Burnaby is an arbitrarily set limit to separate the routes of the mail runs. These two sets of data have no visually detectable clusters (Figure 3 and 4) and the present routes of 5

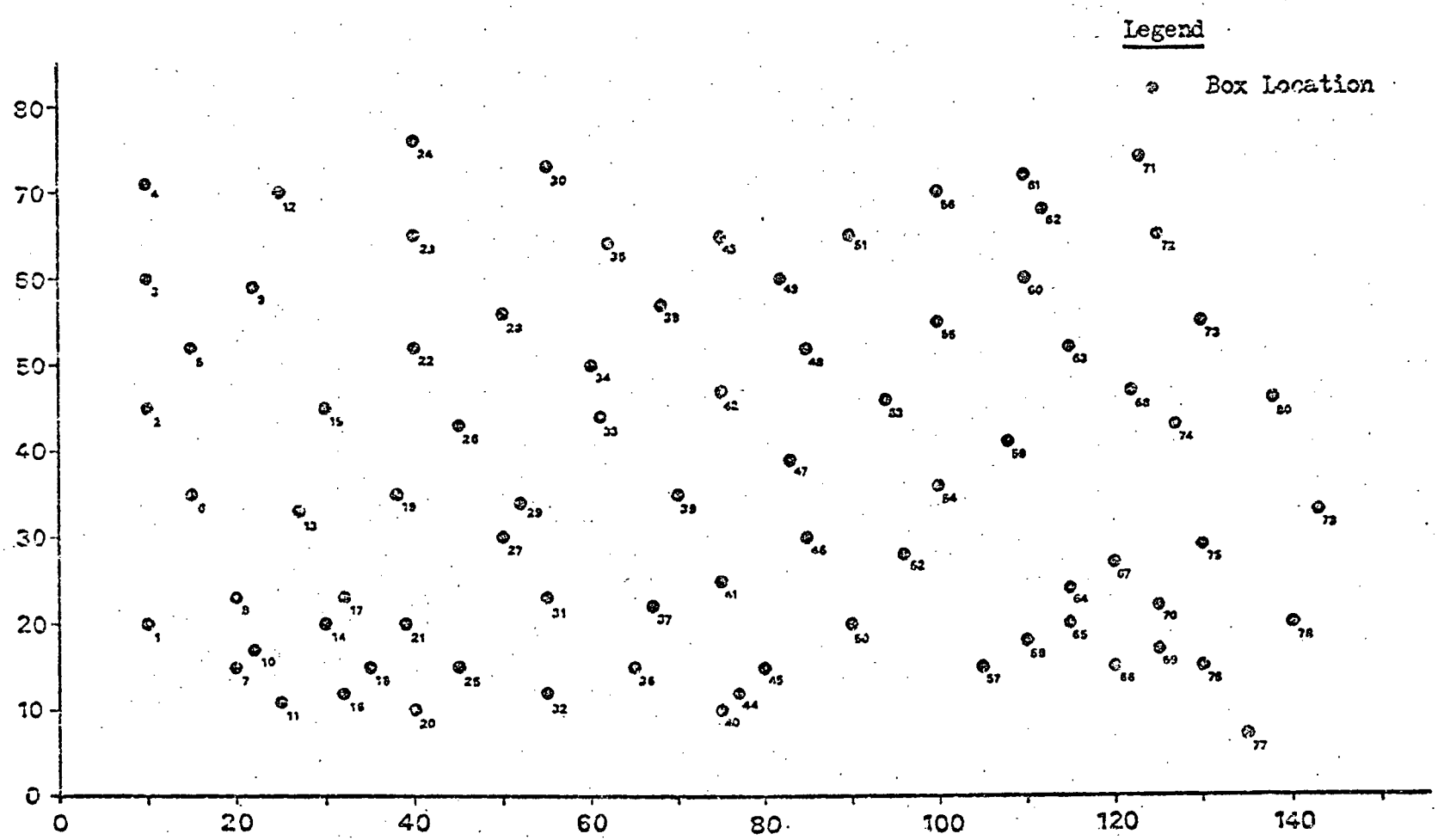


Figure 1. Location Map of Evenly Distributed Contrived Data Set (DATA1)

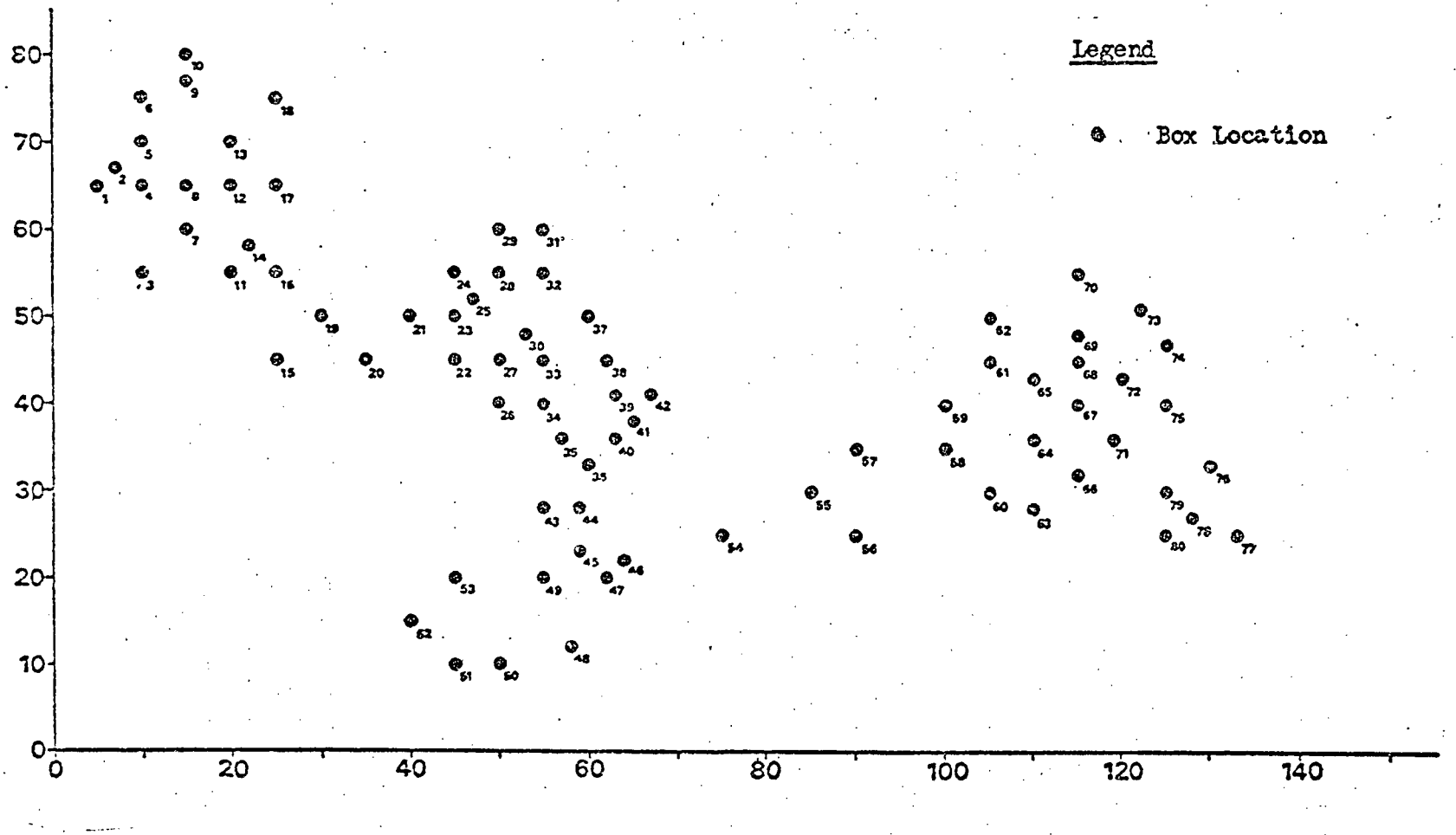


Figure 2. Location Map of Unevenly Distributed Contrived Data Set (DATA2)

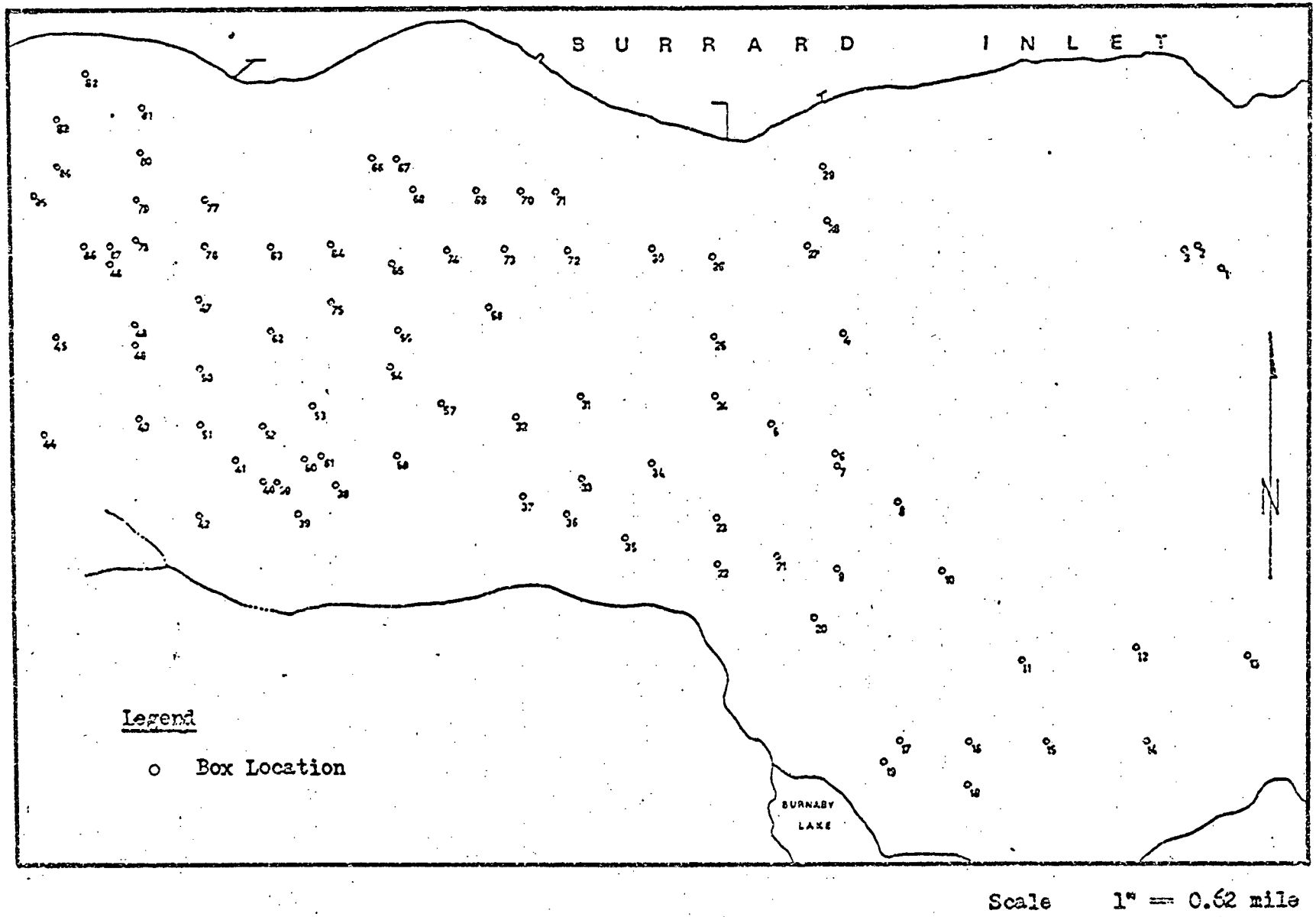


Figure 3. Location Map of North Burnaby Mail Boxes (NBDATA)

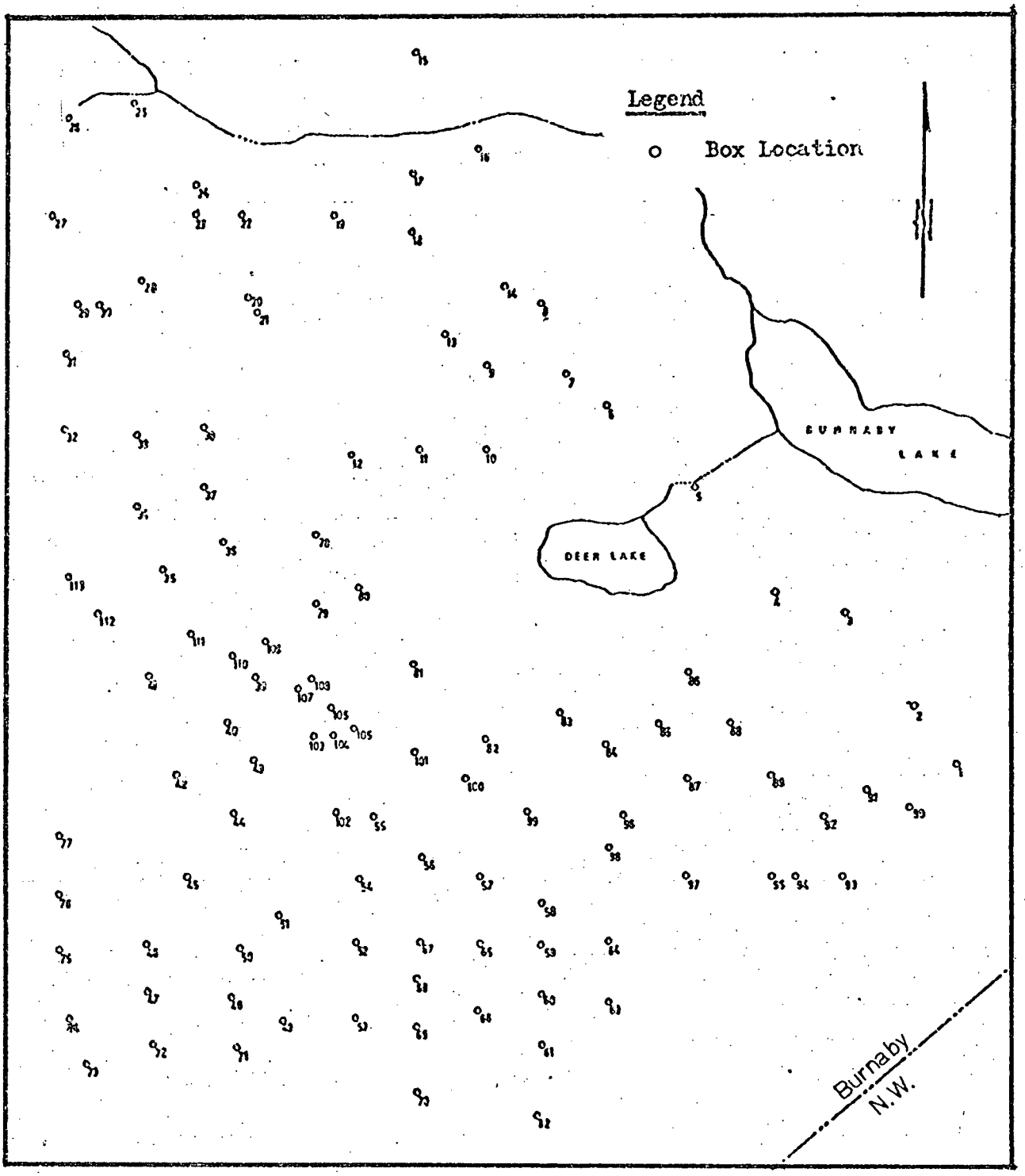


Figure 4. Location Map of South Burnaby Mail Boxes (SBDATA)

trucks serving these areas do not conform to any grouping system. There are 87 and 113 box locations in North and South Burnaby respectively. The x and y-coordinates of these localities are all measured with respect to the grid system adopted by the Post Office.

All four sets of data have their own spatial characteristics. The contrived data sets are designed to test the ability of different clustering techniques to outline visually detectable group boundaries. On the other hand, the empirical data sets are used to examine the validity of the cluster methods as a tool for grouping non-patterned actual locations of boxes for the Post Office. The implications in using comparisons of grouping resulting from different techniques on these four data sets would depict, if any, the "natural" groupings inherent to each of these data sets.

5.1.2 Association Measure

The two-dimensional Cartesian data sets tested in this study are actually localities of data points with two unrelated variables. These mathematically unrelated variables of any given data point in the set have ruled out the use of correlation measure as the clustering criterion. The other alternative is distance measure. Among the various types of distance measures, the Euclidean one is probably the most suitable measure for clustering algorithms.

Traditionally, the true distance between two points i and j can be expressed, in terms of their x and y coordinates, as:-

$$d_{ij} = k((x_j - x_i)^p + (y_j - y_i)^p)^{1/p} \qquad (1)$$

where k and p are derived from analysis of the data set. The value of k and p both reflect the actual distance travelled from point i to point j . A study of the scheduling problem (Tse, 1975) related to the Post Office mail runs indicates that the value of k and p depend on the road pattern and the accessibility from one point to another. It is generally acceptable to use k and p as 1, and formula (1) will become

$$d_{ij} = |x_j - x_i| + |y_j - y_i| . \qquad (2)$$

In this study, the latter form is used because:-

(1) the two axes of the city road grid are mostly rectangular to each other;

(2) most mail boxes are located at the corner of a street where the roads intersect; and

(3) the travelling distance from one location to another would be the summation of horizontal distance along the E-W direction and the distance along the N-S direction. This distance measure is referred to as "City-Block" distance in this study.

5.1.3 Inputs to Clustering Methods

As described in Chapters III and IV, the inputs required by various hierarchical and nonhierarchical differ in many aspects. Basically, similarity matrix and data variables are the two major inputs. In this study, stored matrix approach is used for six hierarchical methods, and data variables are inputs to the other six clustering runs.

The elements of the symmetric similarity matrices containing distance measures of the four data sets are computed by a small computer program, MATRIX (Appendix A). This program stores the values of the variables of each data point in a set in the computer memory, and distances, d_{ij}, from point i to j are computed using formula (2) in the previous section. These distances are later scaled to values less than 1.0 as required by some of the clustering programs. The elements of the lower triangle of the symmetric matrix are then sorted in rows of 10 elements into a file for clustering procedures.

Raw data variables are all punched on cards in predetermined formats for various clustering algorithms. All these variables are in units of millimeter conforming to the system used by the Post Office.

5.1.4 The Number of Clusters

The number of clusters is an important key in the nonhierarchical methods used in this study. The four data sets are grouped into either two or three groups. The number of clusters for both contrived data sets are arbitrarily set: 2 for the evenly distributed data set and 3 for the other. On the contrary, the number of clusters for North and South Burnaby areas are predetermined: there are 2 and 3 routes for North and South Burnaby respectively, and it is only logical to use 2 and 3 clusters accordingly.

There is no requirement to define the number of clusters for hierarchical methods. These methods group data sets into one single cluster at the final stage, thus it is not necessary to predifine the number of groups. In order to compare results of both hierarchical and nonhierarchical methods, the number of clusters for each data set are correspondingly equal as stated in the above paragraph. The groups for hierarchical methods are identified in the tree diagram representing the dendogram of the clustering results.

5.1.5 What to Cluster

Obviously there is no choice of what to cluster. The unanimous key for cluster, in this case, is the data unit. The two totally unrelated variables would make the clustering

of variables unfeasible and meaningless. The independence of the variables among data points further discourage the use of variables as the basic cluster unit.

5.1.6 Clustering Techniques

The nine hierarchical and three nonhierarchical clustering methods described in Chapters III and IV are the 12 better known ones and are probably most applicable to distance measures than other complex computational methods. In this study, various computer programs for these clustering methods were used.

Fortran programs for single linkage (using minimum Euclidean distance as criterion), complete linkage, average linkage within new group, average linkage between merged groups, centroid and median methods are all modified from Anderberg's (1973) Appendix E (pp. 275-305). The inputs for these programs include the number of entities and the lower triangle of the symmetric similarity matrix. This set of programs is made up of the main program DRIVER; subroutines CNTRL, CLSTR, MTXIN, TREE, and METHOD; and function LFIND (Appendix B). The use of different versions of METHOD would generate results for the above six methods. The result of each cluster procedure is presented in a horizontal tree diagram (Appendix B). From these tree diagrams, groups are identified and plotted accordingly onto figures.

The UBC:BMDP2M program is basically a single linkage clustering routine. In this program, Engelman and Fu (1970) use either square roots of the sum of squares of differences (Euclidean distance) or chi-square of the data points as distance measure. Both these criteria give drastically different results from that of single linkage using simple Euclidean distance as a criterion measure. Data variables are the inputs to this program, and a vertical tree diagram (Appendix C) recording the clustering sequences is output from this computer package program.

Another UBC package program CGROUP (Patterson and Whitaker, 1973) uses Ward's error sum of squares grouping techniques to cluster data points with variables. Similar to BMDP2M program, the inputs to CGROUP include the set of variables for each data unit and the options for running the program as well as outputing the results. The output contains a detailed sequence of the clustering procedure, the group membership at each step, and a vertical tree diagram. An optional output is the plot of the error sum of squares versus the number of groups (Appendix D).

A program for three hierarchical methods are modified from Anderberg's (1973) Appendix F(pp. 306-325). This program is designed to implement the three nearest centroid sorting techniques described in Chapter IV. The Forgy's and Jancey's grouping methods are options in a version of

subroutine KMEAN, and the Convergent K-Mean method is implemented in another version of KMEAN. The whole program is composed of DRIVER, the main program; and 3 subroutines: EXEC, RESULT and KMEAN (Appendix E). Either seed points or initial partitions can be used to initiate the clusters in this program. The other inputs include the number of entities, the number of variables for each entities, the number of clusters for this set of data, optional output features and the actual variables for each data point. The output is essentially a list of membership within each resulting cluster. The number of entities moved in the iterative steps is also ouput (Appendix E).

All the above programs are used to generate different outputs for this study. The single linkage methods (Euclidean, sum of square of differences and chi-squares) are used to test the effect of measures on grouping results. Other methods are just straight forward implementation of the other nine methods described in Chapters III and IV. The results from these trials are then compared and evaluations of these techniques are discussed in the following sections.

5.2 Tool for Interpretation of Results

The outputs from all the computer programs give various representations of the clustering results. Tree diagram together with the clustering sequence are commonly

the hierarchical outputs. On the other hand, only a membership list is output from nonhierarchical methods. This inconsistent representation of outputs presents a problem in comparing the results effectively.

Tree diagram is actually a very effective tool for interpreting the clustering results. However, if there are more than 50 points in the data set, the tree becomes complex and it is difficult to trace the tree without a step by step follow-up of the sequence at the same time.

Membership list is not useful at all as a tool for interpretation unless frequent referrals to the data unit inputs are made. Multi-dimensional data are therefore very difficult to interpret if the representation space is more than 3-dimensions. The 2-dimensional data used in this case, however, can be plotted onto maps according to the values of the data variables. The results from both hierarchical and non-hierarchical methods - the linking of entities and the grouping lists, can be plotted onto maps of the data units. These plots of the results are the keys to interpretation and comparisons.

5.3 Results

The results from different clustering techniques from the computer programs are plotted onto maps for comparisons. The links of entities are plotted as straight lines

between points in the graph. The sequential links of all the points as output by the hierarchical programs are charted using the lowest indexed point as the link to another entity or group lead point (the lowest indexed point of the group). These results in diagrams of many, if not confusing, links between data points. The recognization of the last few links among groups or entities in the diagrams allow the user to identify the group boundaries fairly easily.

The group boundary for nonhierarchical results are easier to handle. The data point index on the graph is identical to that on the membership output list, thus boundaries of the clusters can be easily plotted onto the diagram.

In view that the four sets of data have different spatical characteristics, the results of the 12 methods of each data set are presented in the following sub-sections for easy identification. The discussion of results is also included in these sub-sections.

5.3.1 Evenly Distributed Contrived Data (DATA1)

A total of 80 data points exists in this data set (Figure 1 and Appendix F). Different randomly chosen initial seed points and initial partitions were input for the three nonhierarchical methods, and the resulting groupings are identical in all these trials (Table 3, Figures 14 and 15).

Group Sizes

Trial	1		2		3	
Group	1	2	1	2	1	2
Methods \ Seed Points	26	59	20	66	15	50
Jancey's	36	44	44	36	36	44
Forgy's	36	44	44	36	36	44
Convergent K-mean	36	44	44	36	36	44
Methods \ Initial Partition	40	40	35	45	30	50
Jancey's	44	36	36	44	36	44
Forgy's	44	36	36	44	36	44
Convergent K-mean	44	36	36	44	36	44

Table 3. Summary of Nonhierarchical Runs for DATA1

The results from different clustering techniques are plotted as shown in Figures 5 to 15. Undoubtly the different methods used give various results in the number of entities within the groups and membership list of these groups. Table 4 summarizes the number of entities for the two defined groups for this data set. The lists of memberships of the two groups resulting from each method are included in Appendix G.

The results presented in the diagrams and tables for this evenly distributed contrived data set give various group sizes as well as group memberships. These differences in results are the end products of different clustering criteria and algorithms. Among the hierarchical methods, the number of entities in group 1 and 2 is perhaps best balanced in the Ward's method (group sizes of 43 and 37 respectively). This, however, does not necessarily indicate that Ward's error sum of squares method is the best for grouping evenly distributed data set. The approach in evaluating the results in section 5.5 gives a more comprehensive judgement on the superiority of different clustering methods. It is interesting to notice that the Centroid and Median methods both have the same clustering sequences and group members in the clusters. Slight variations of the group members in the two average linkage methods reflect the similarity of these two algorithms. The same applies to the single linkage methods using "City-Block" and Euclidean distance as measures. Chi-square

Group	Group Sizes 1	2
Methods		
Hierarchical		
Single Linkage		
City - Block	67	13
Euclidean Distance	67	13
Chi-Squares	22	58
Complete Linkage	35	45
Avg. Linkage between Merged Group	27	53
Avg. Linkage within New Group	27	53
Centroid Method	49	31
Median Method	49	31
Ward's Method	43	37
Nonhierarchical		
Jancey's Method	36	44
Forgy's Method	36	44
Convergent K-mean Method	36	44

Table 4. Results of 12 Clustering Methods for DATA1

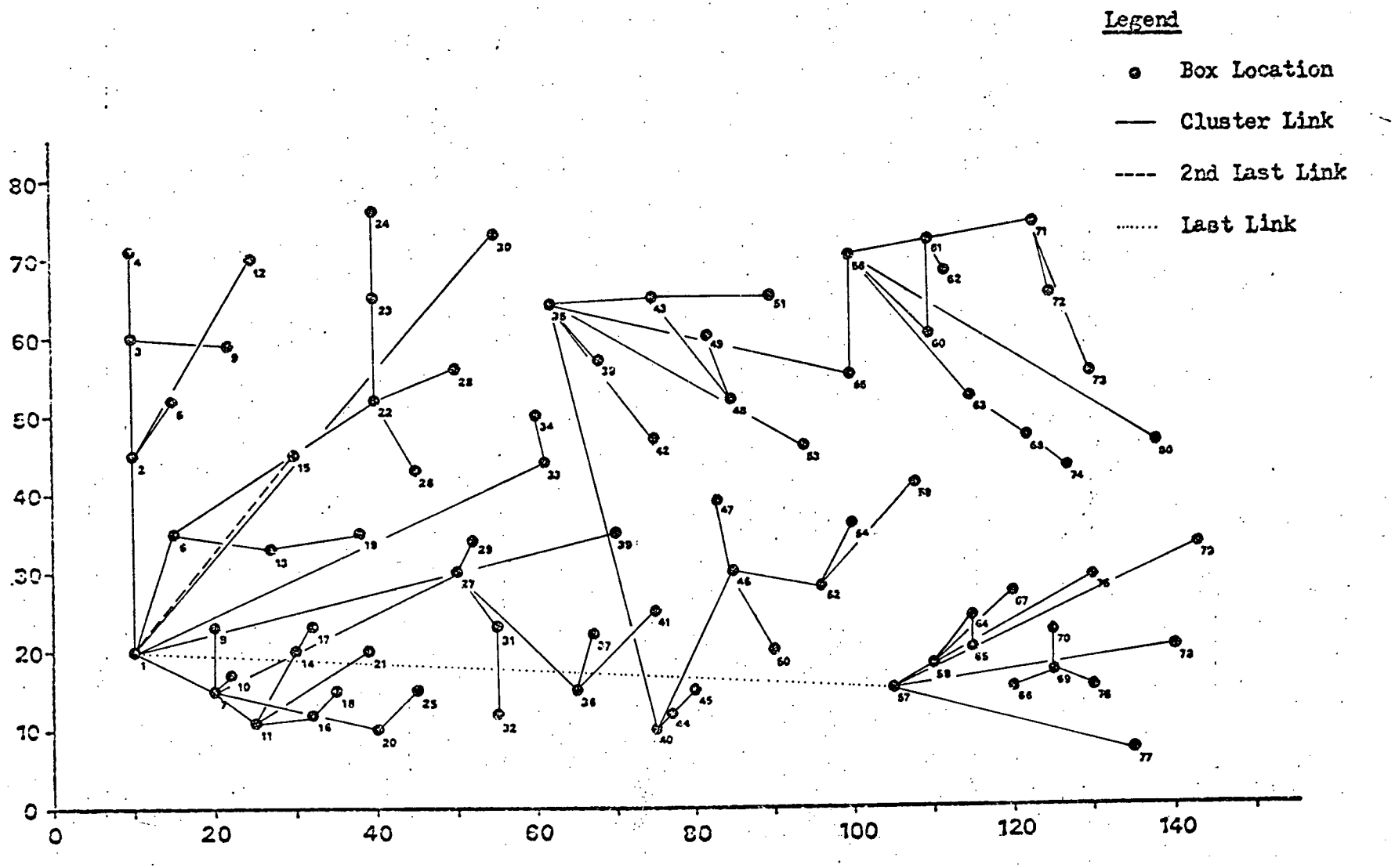


Figure 5. Linkages Outlined by Single Linkage - " City - Block " Method for DATA1

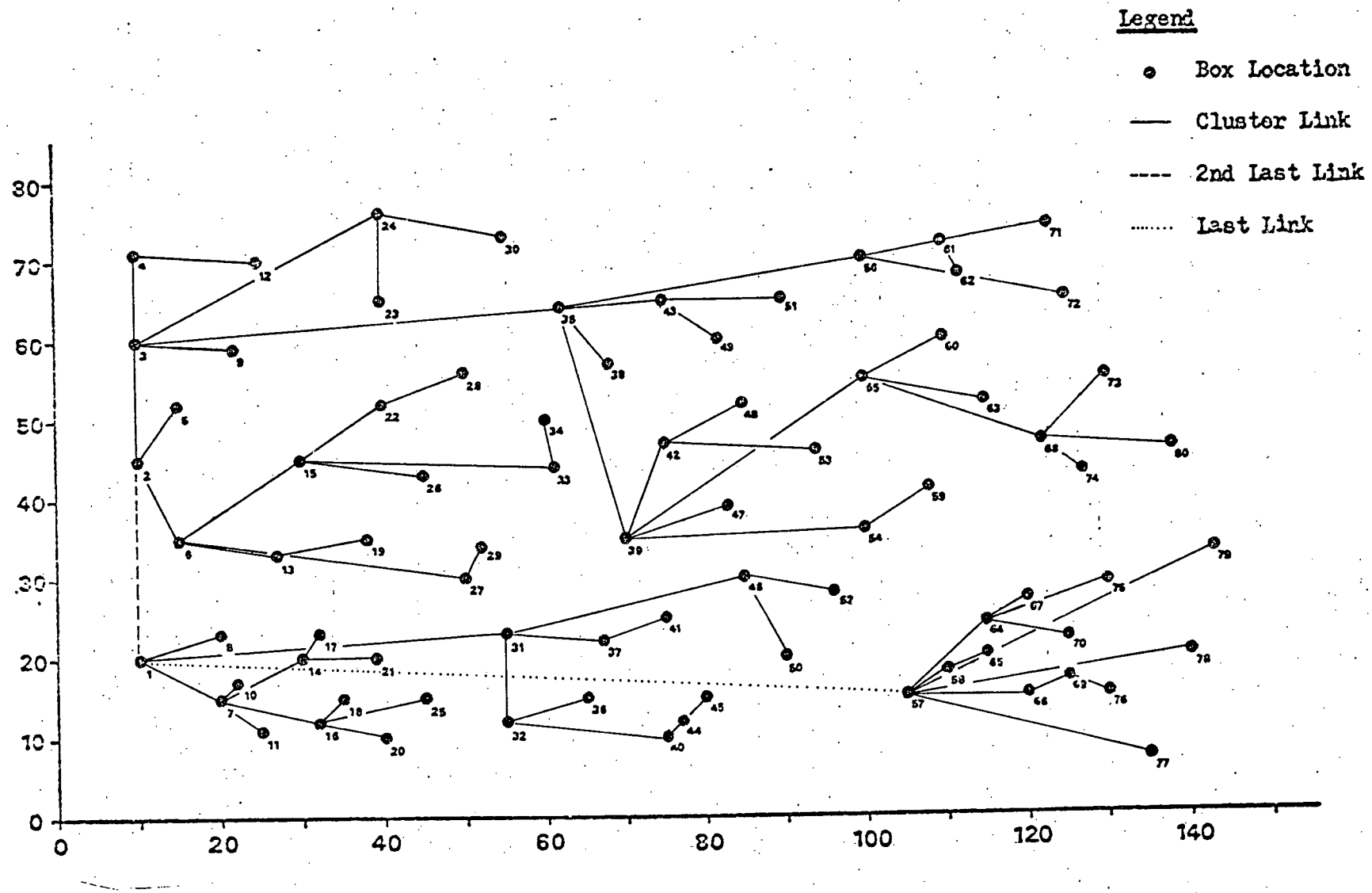


Figure 6. Linkages Outlined by Single Linkage -Euclidean Distance Method for DATA1

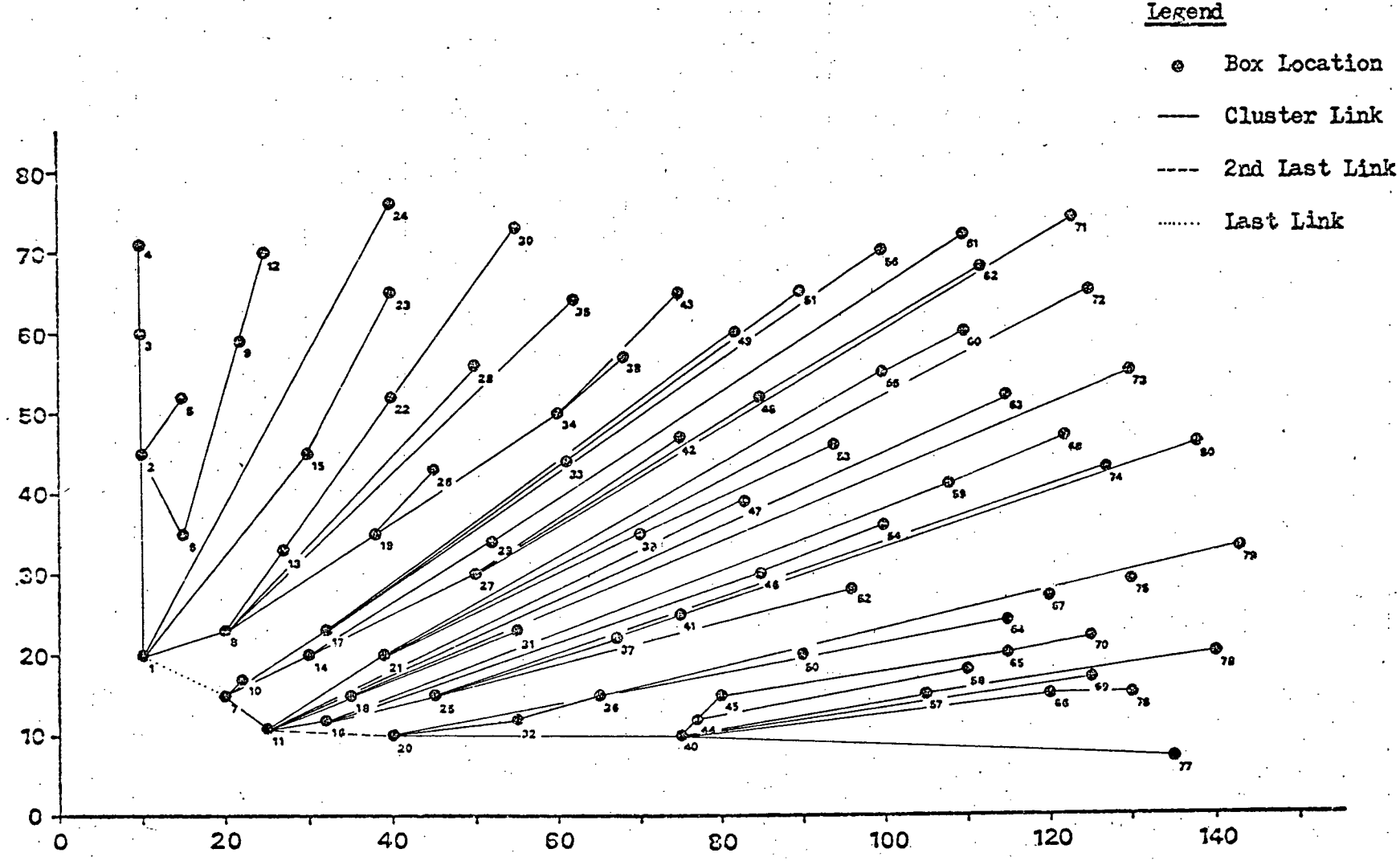


Figure 7. Linkages Outlined by Single Linkage-Chi Squares Method for DATA1

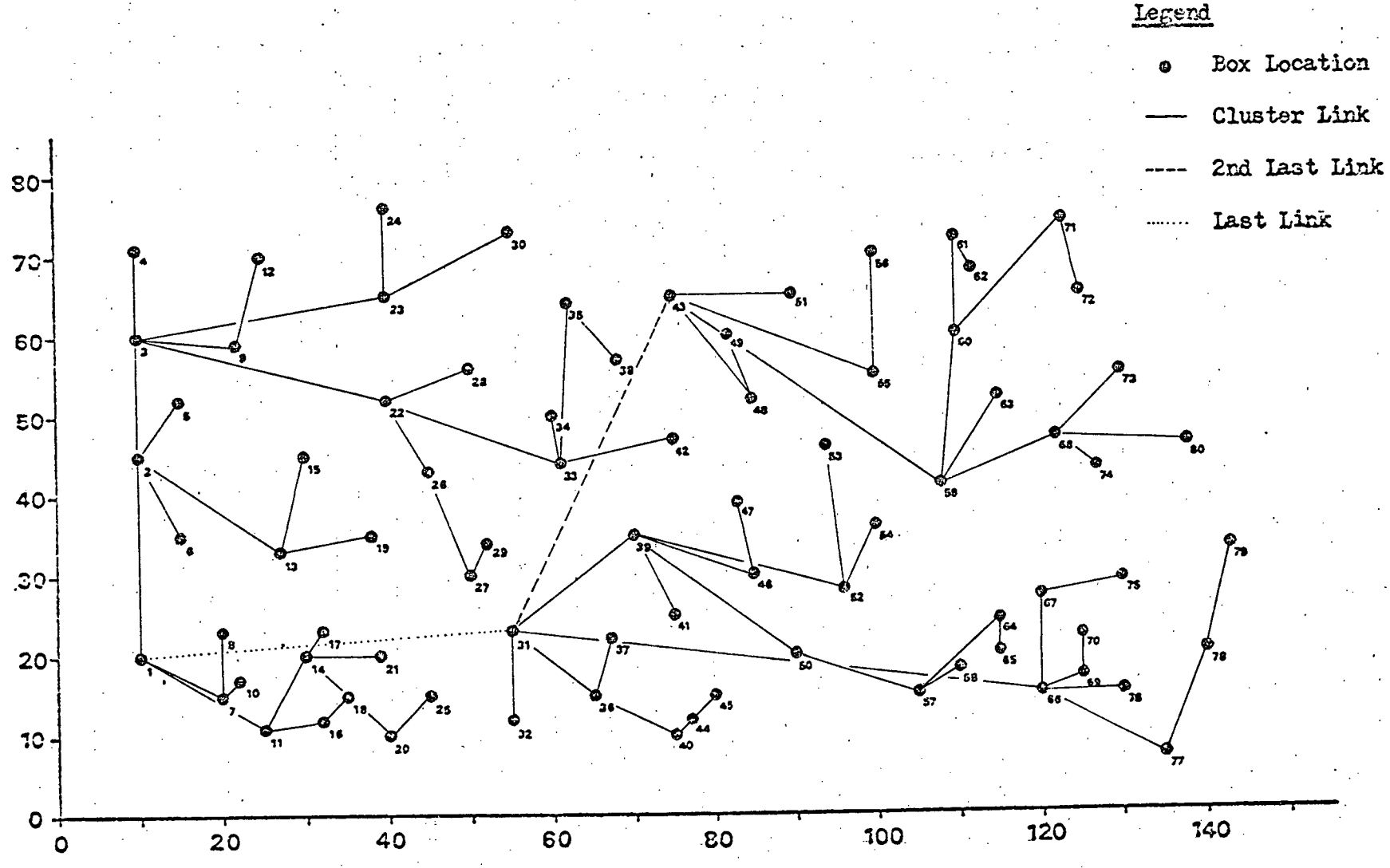


Figure 8. Linkages Outlined by Complete Linkage Method for DATA1

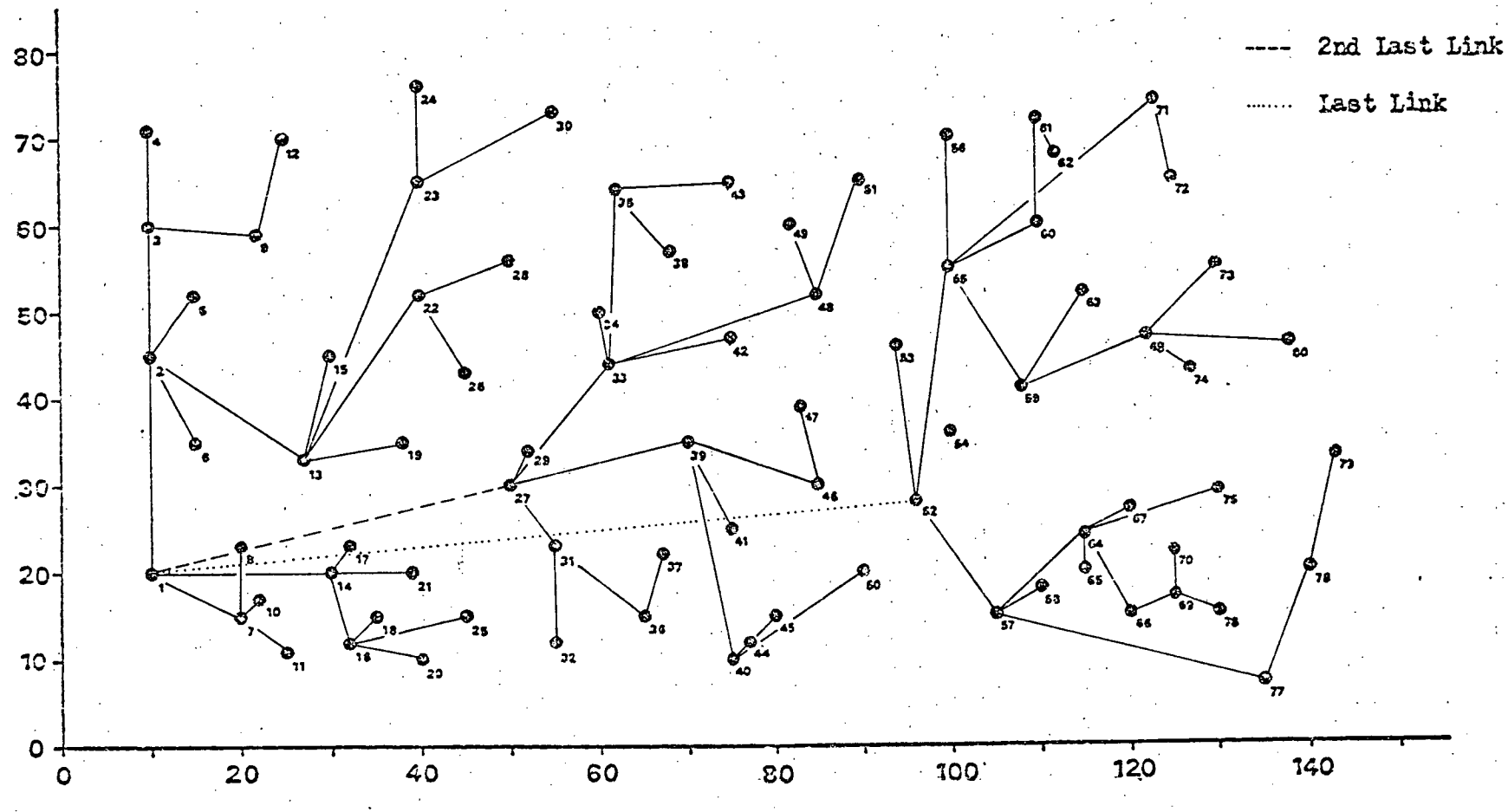


Figure 9. Linkages Outlined by Avg. Linkage between Merged Groups Method for DATA1

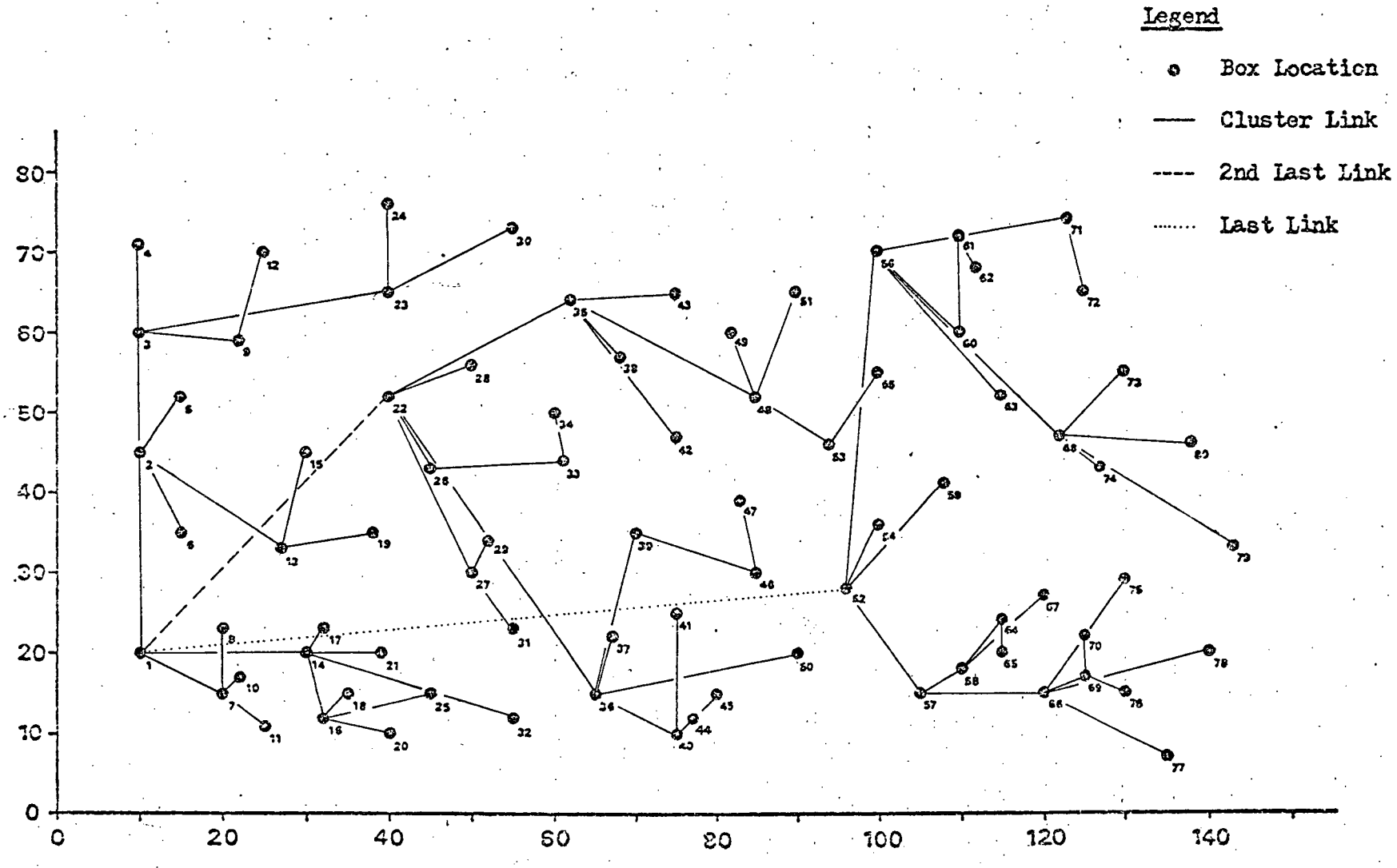


Figure 10. Linkages Outlined by Avg. Linkage within New Group Method for DATA1

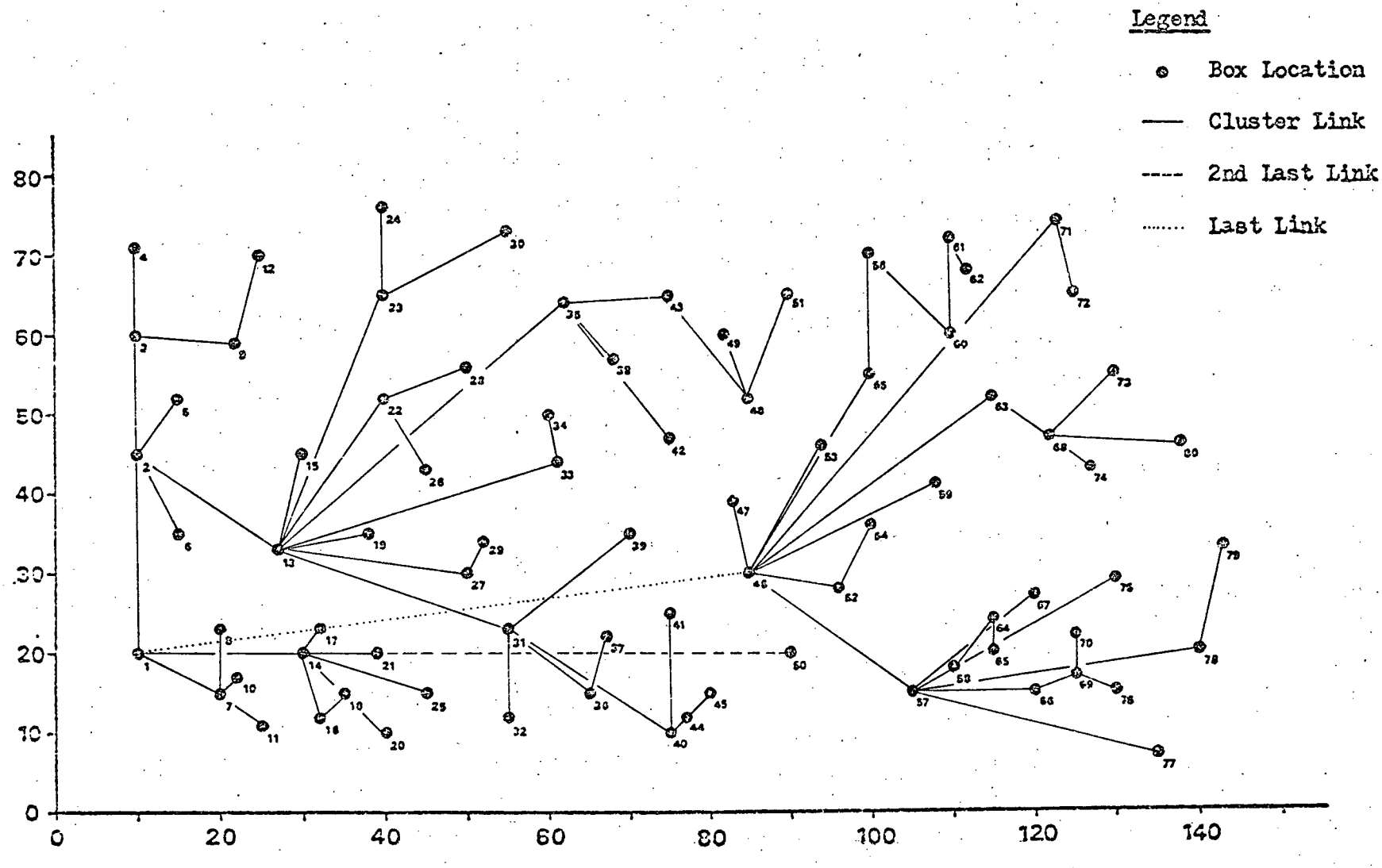


Figure 11. Linkages Outlined by Centroid Method for DATA1

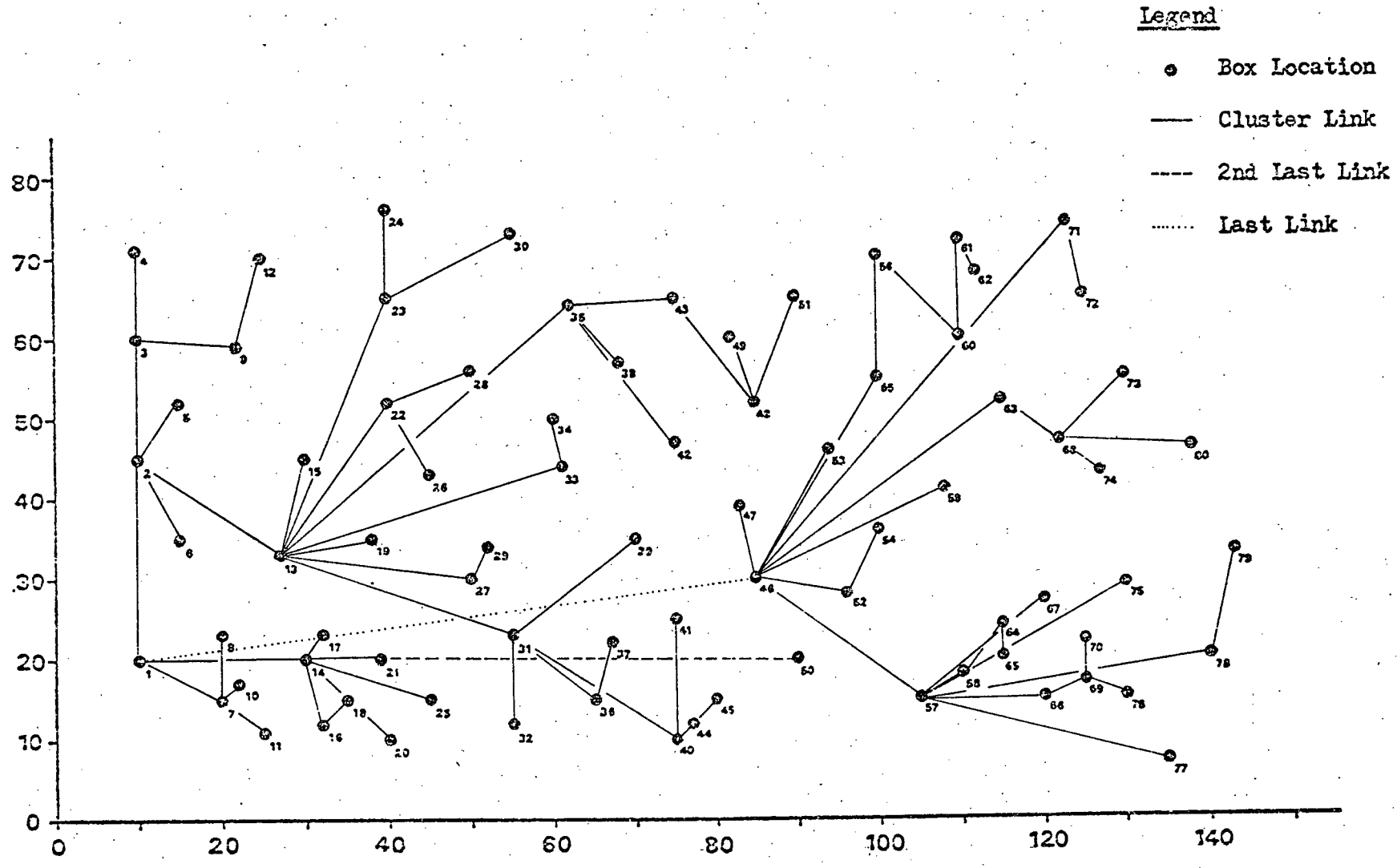


Figure 12. Linkages Outlined by Median Method for DATA1

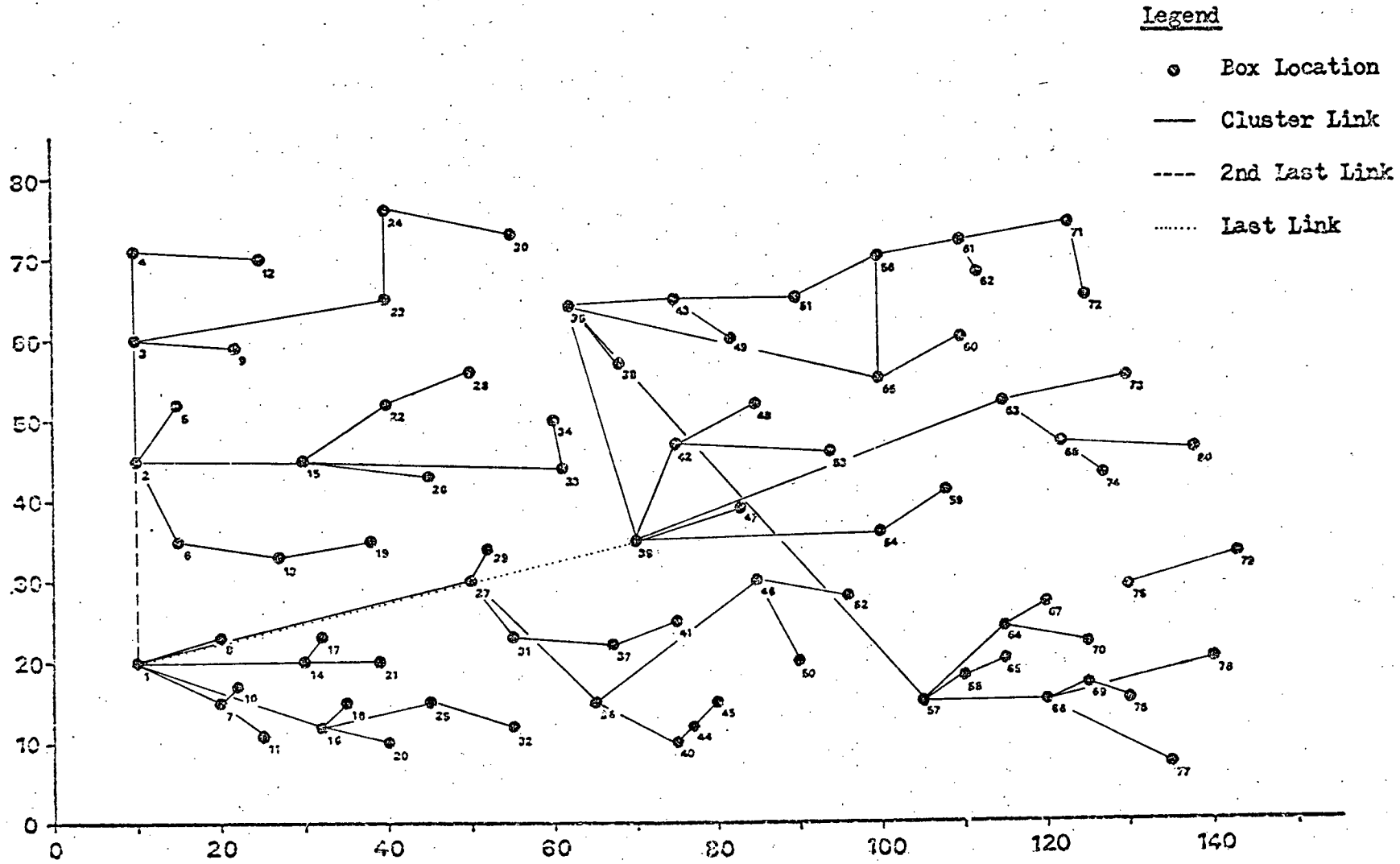


Figure 13. Linkages Outlined by Ward's Method for DATA1

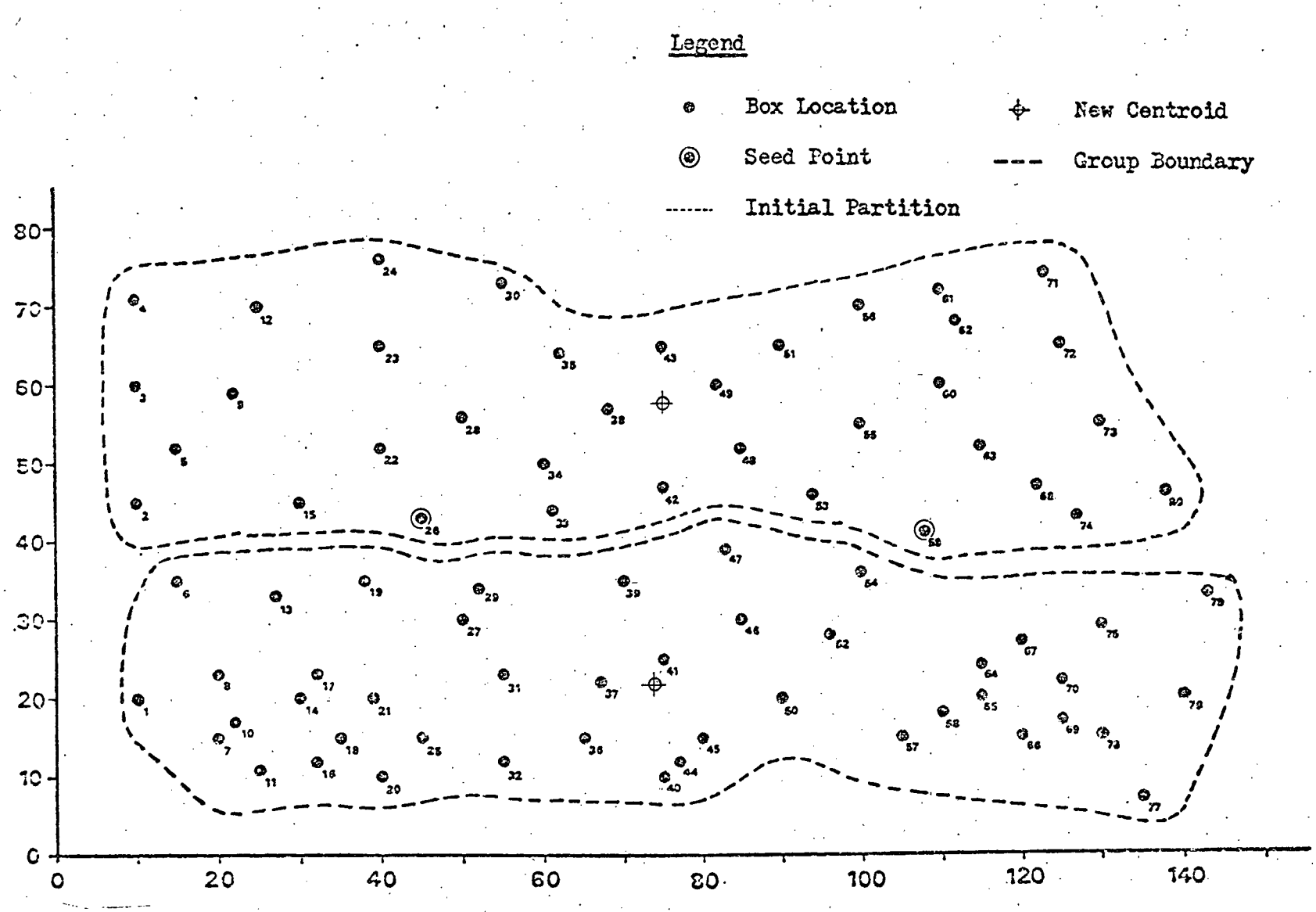


Figure 14. Group Boundaries Defined by 3 Nonhierarchical Methods Using Seed Points as Inputs for DATA1

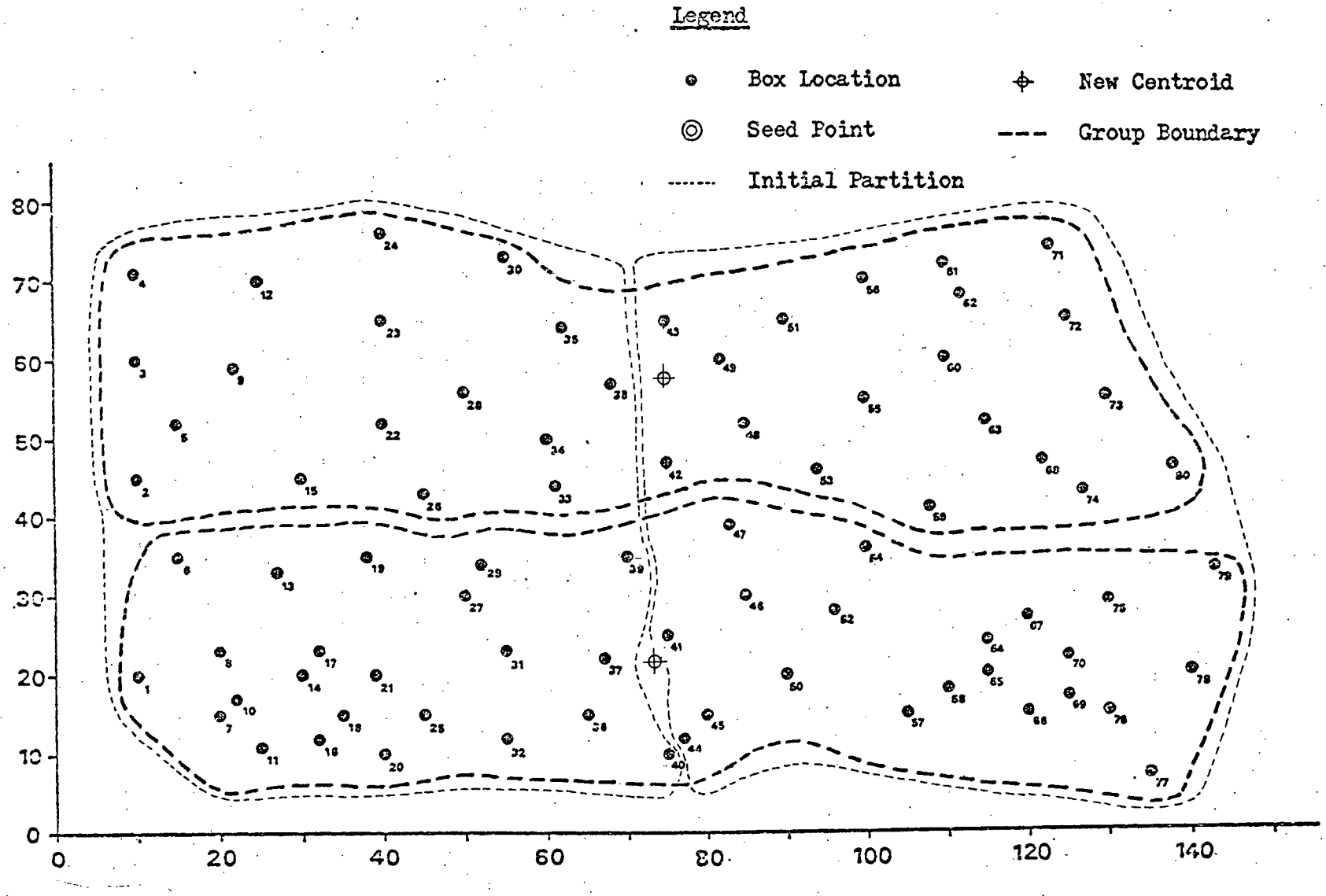


Figure 15. Group Boundaries Defined by 3 Nonhierarchical Methods Using Initial Partitions for DATA1

single linkage method results, however, do not resemble any of the hierarchical method, and it is doubtful that this method is suitable for clustering this evenly distributed data set.

All the nonhierarchical results are identical and they are fairly evenly balanced in group sizes. These non-hierarchical methods have a unique characteristic different from the hierarchical ones: the clusters are grouped laterally instead of radially as defined by most hierarchical methods. This characteristic would definitely affect the within group inter-unit travel times and distances, also the distribution of distances between units would be different from that of the other methods. These results, however, do not indicate the superiority of one class of method over the other. Further evaluation of these methods is included in section 5.5.

5.3.2 Unevenly Distributed Contrived Data (DATA2)

This data set has 80 data units located in 3 visually distinguishable groupings (Figure 2 and Appendix F). Similar to the trials carried out for the evenly distributed data set, trials of using different seed points and initial partitions for nonhierarchical methods were conducted. The results of these trials differ slightly in using different seed points or initial partitions (Table 5 and Figures 25, 26). This is probably the result of the inclusion of in-between data points

Group Sizes

Trial	1			2			3		
Group	1	2	3	1	2	3	1	2	3
Seed Points / Methods	8	35	67	10	40	60	7	30	70
Jancey's	16	25	39	23	34	23	23	23	34
Forgy's	23	29	28	16	41	23	16	29	35
Convergent K-mean	23	23	34	16	41	23	16	29	35
Initial Partition / Methods	20	34	26	27	27	26	23	23	34
Jancey's	15	25	39	23	23	34	16	39	25
Forgy's	23	23	34	23	23	34	23	34	23
Convergent K-mean	23	23	34	23	23	34	23	34	23

Table 5. Summary of Nonhierarchical Runs for DATA2

for the three visually identifiable clusters. The variance of group sizes is relatively small but this indicates the importance of initial seed points or centroid in using nonhierarchical methods. Resembling the results for DATA1, the clusters defined by these methods extend laterally instead of radially as in hierarchical methods.

The results from none hierarchical methods differ greatly from one another (Table 6 and Figures 16-24). In this set of results, the Centroid and Median methods have identical clustering sequences as well as membership. Contrary to results for the evenly distributed data set, single linkage method using " City - Block " measure does not resemble any other linkage methods or clustering methods. The existence of 1-member group for the 3 clusters in this method urges to draw a conclusion that single linkage - "City-Block" method is inappropiate for grouping unevenly distributed data set. The other two single linkage approaches, on the other hand, give identical memberships to the 3 groups though sequences of grouping are different. The average linkage methods also give identical memberships as well as grouping sequences to the 3 groups. The intended group sizes are 19, 34 and 27 and the results from complete linkage method are identical to this predetermined group sizes. The average linkage methods also give very similar results to these group sizes.

	Group Sizes		
Group	1	2	3
Methods			
Hierarchical			
Single Linkage			
City - Block	53	1	26
Euclidean Distance	17	25	38
Chi-Squares	17	25	38
Complete Linkage	19	34	27
Avg. Linkage between Merged Group	20	33	27
Avg. Linkage within New Group	20	33	27
Centroid Method	20	37	23
Median Method	20	37	23
Ward's Method	42	14	24
Nonhierarchical			
Jancey's Method	23	23	34
Forgy's Method	16	29	35
Convergent K-mean Method	16	29	35

Table 6. Results of 12 Clustering Methods for DATA2

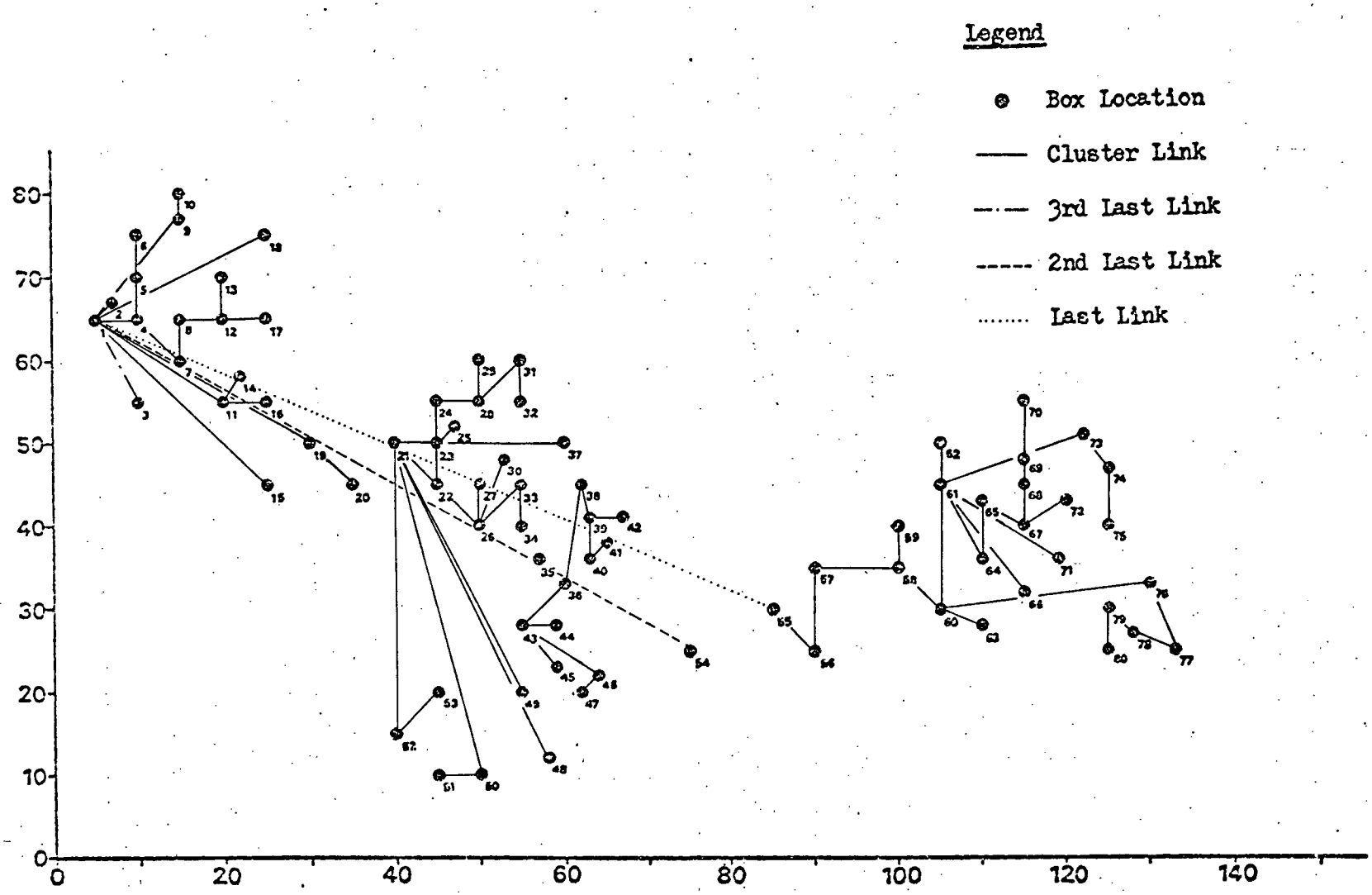


Figure 16. Linkages Outlined by Single Linkage-" City - Block " Method for DATA2

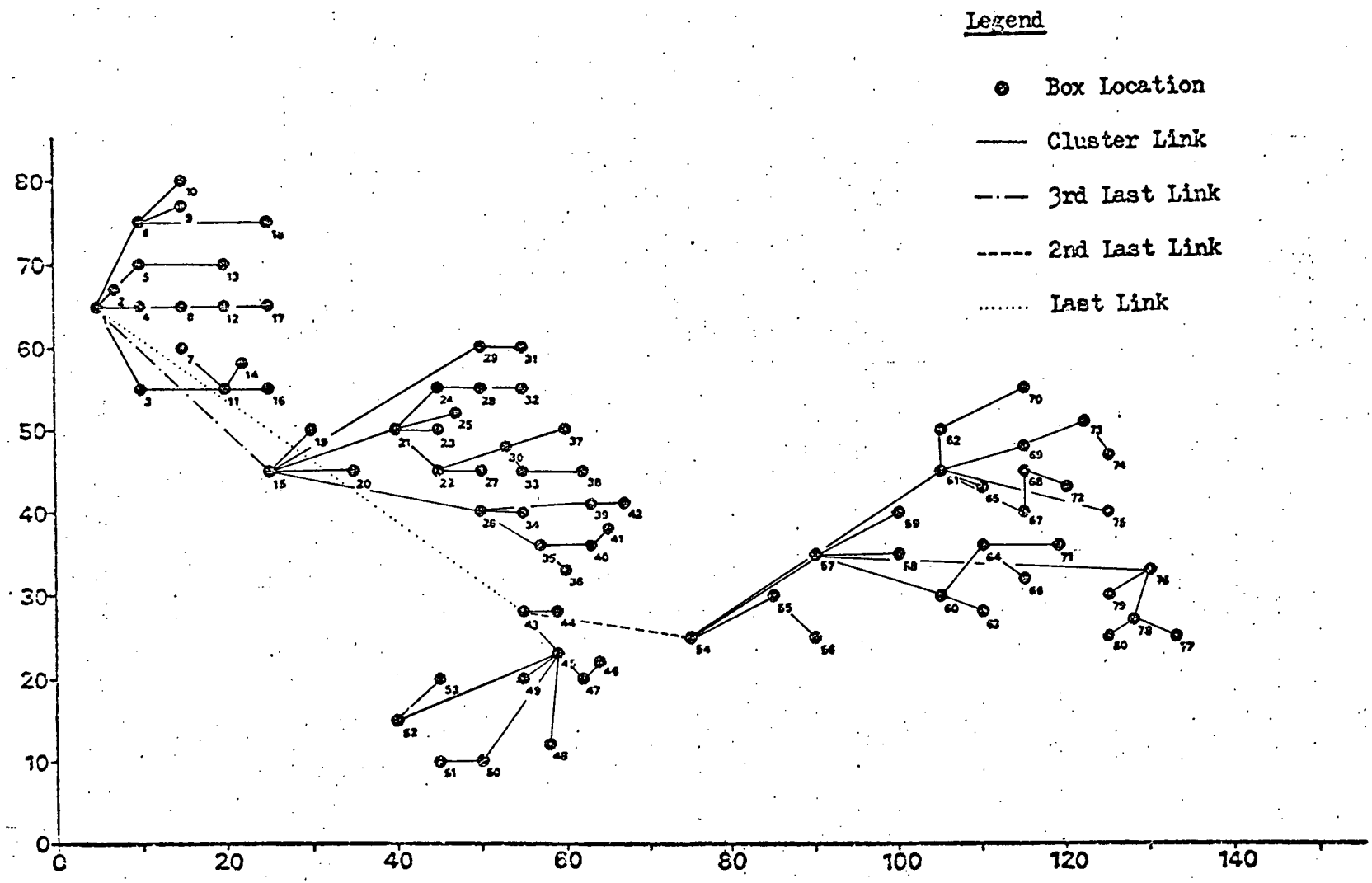


Figure 17. Linkages Outlined by Single Linkage-Euclidean Distance Method for DATA2

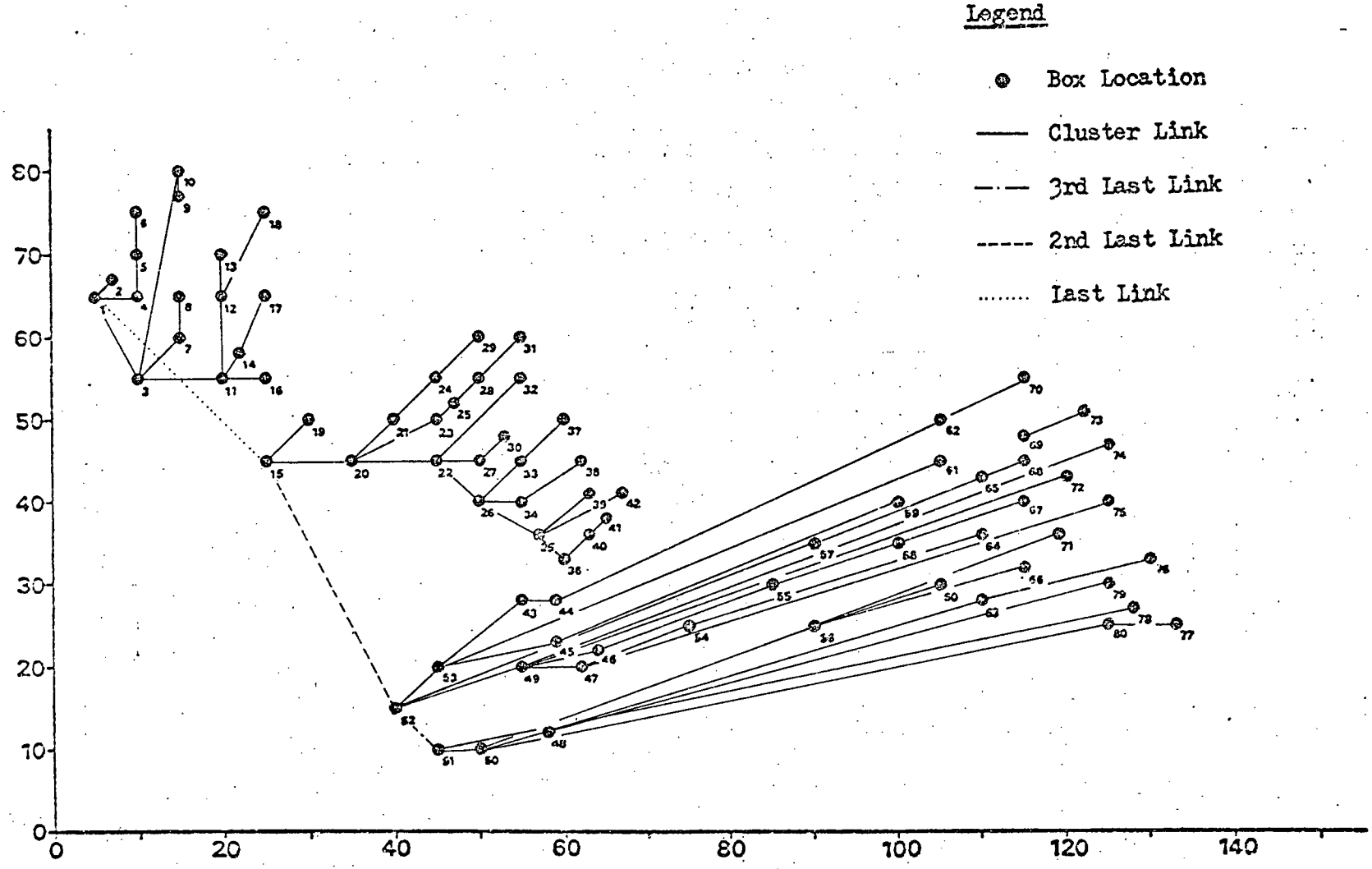


Figure 18. Linkages Outlined by Single Linkage-Chi Squares Method for DATA2

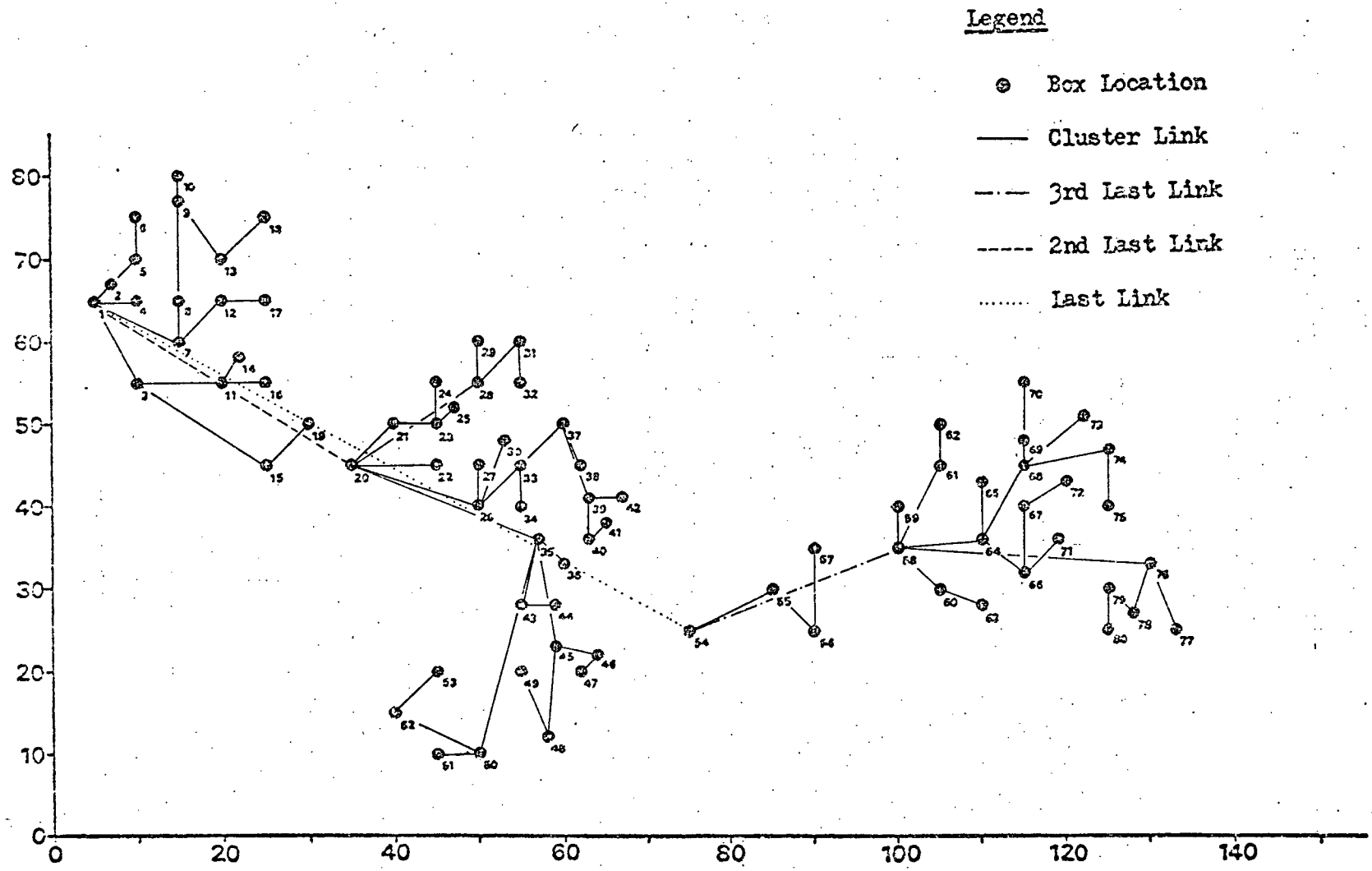


Figure 19. Linkages Outlined by Complete Linkage Method for DATA2

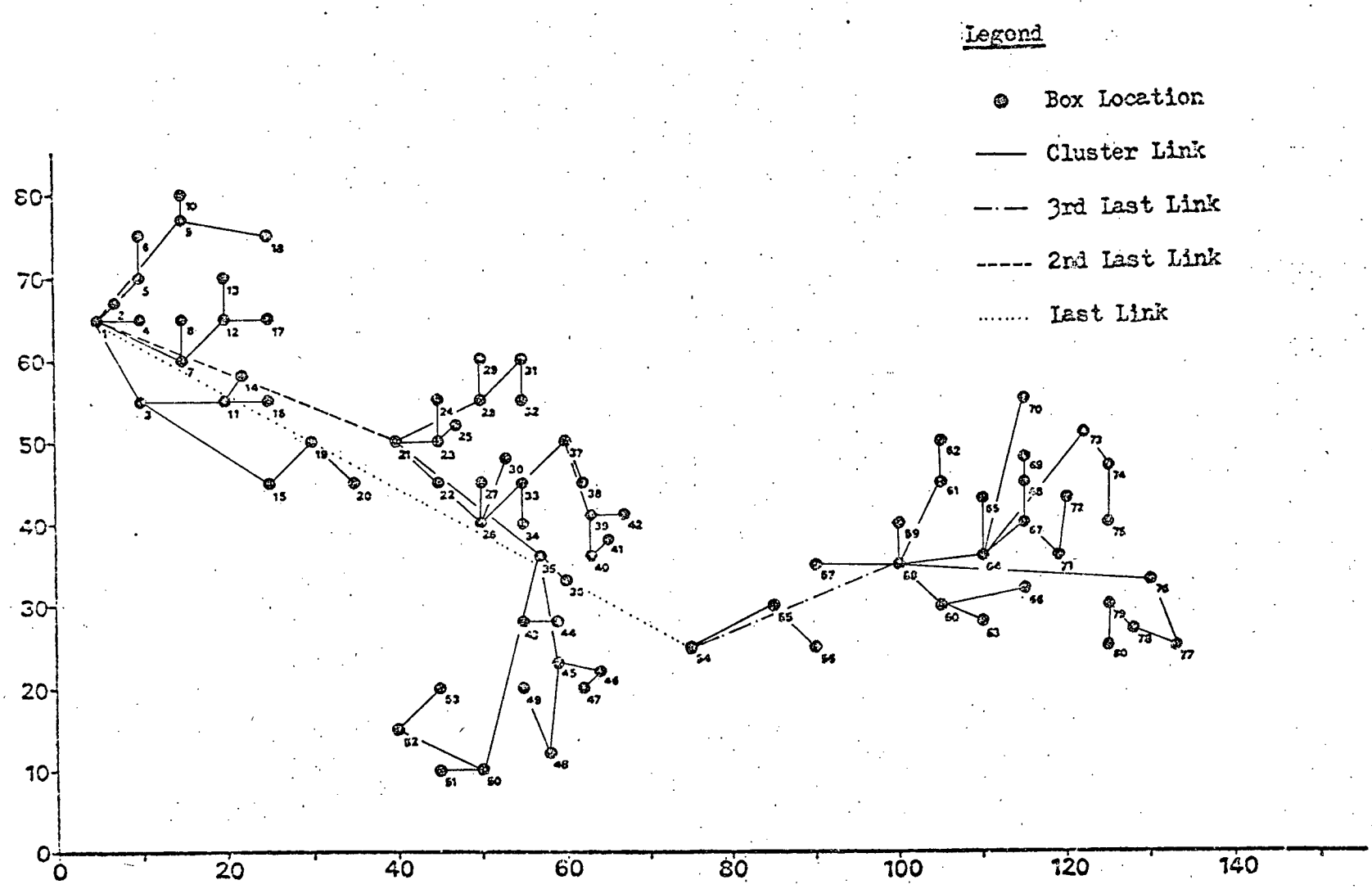


Figure 20. Linkages Outlined by Avg. Linkage between Merged Groups Method for DATA2

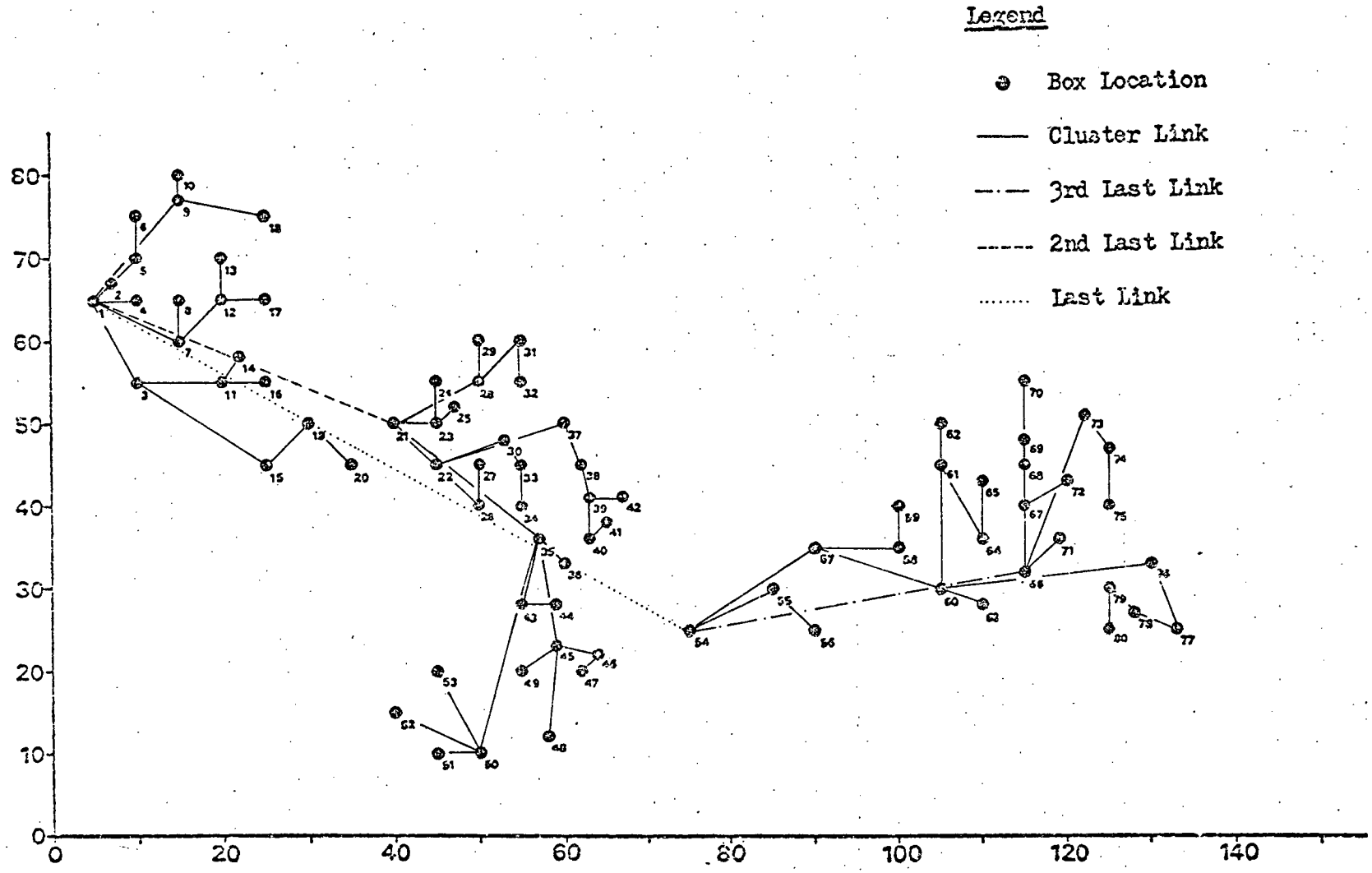


Figure 21. Linkages Outlined by Avg. Linkage within New Group Method for DATA2

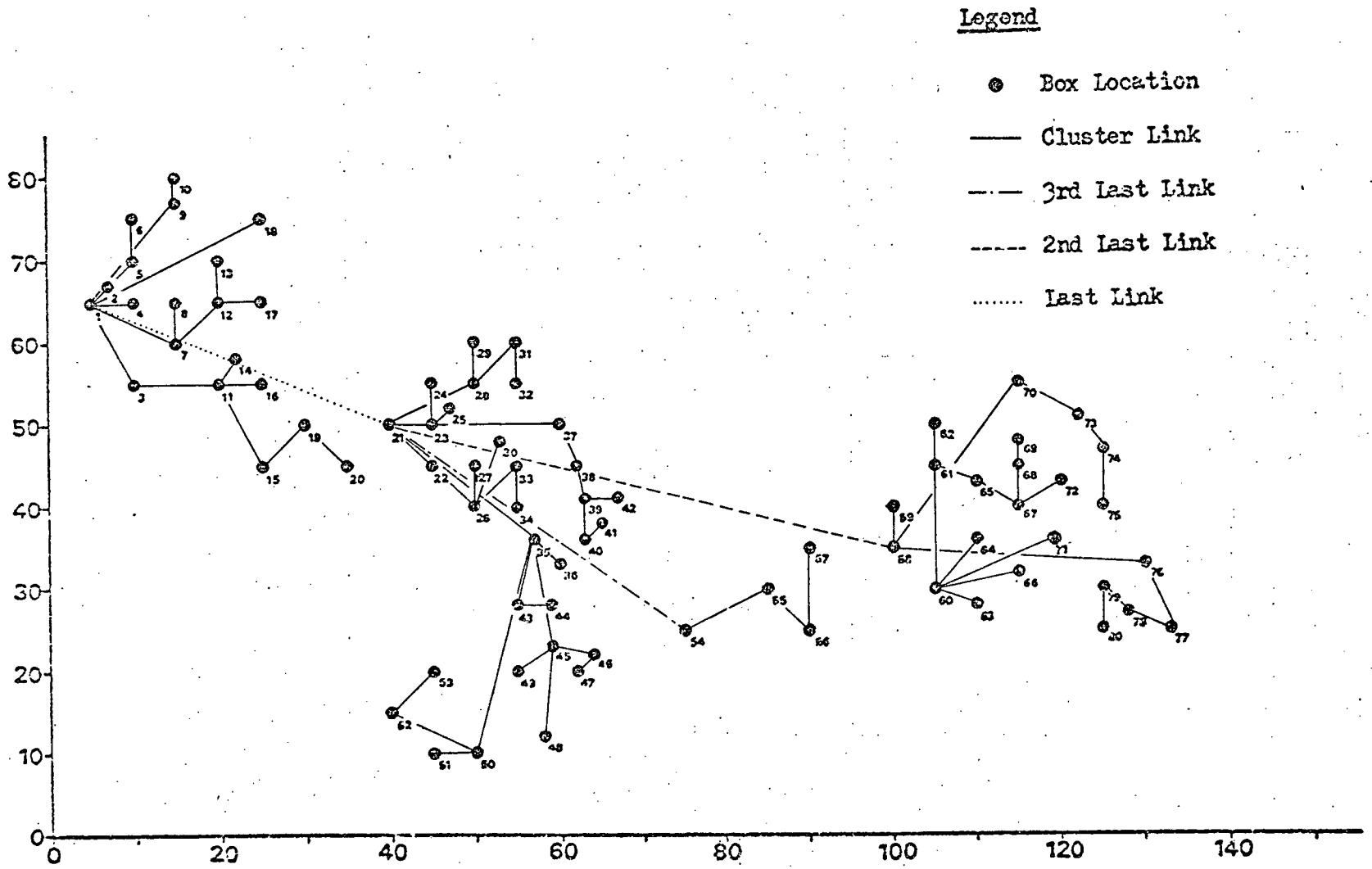


Figure 22. Linkages Outlined By Centroid Method for DATA2

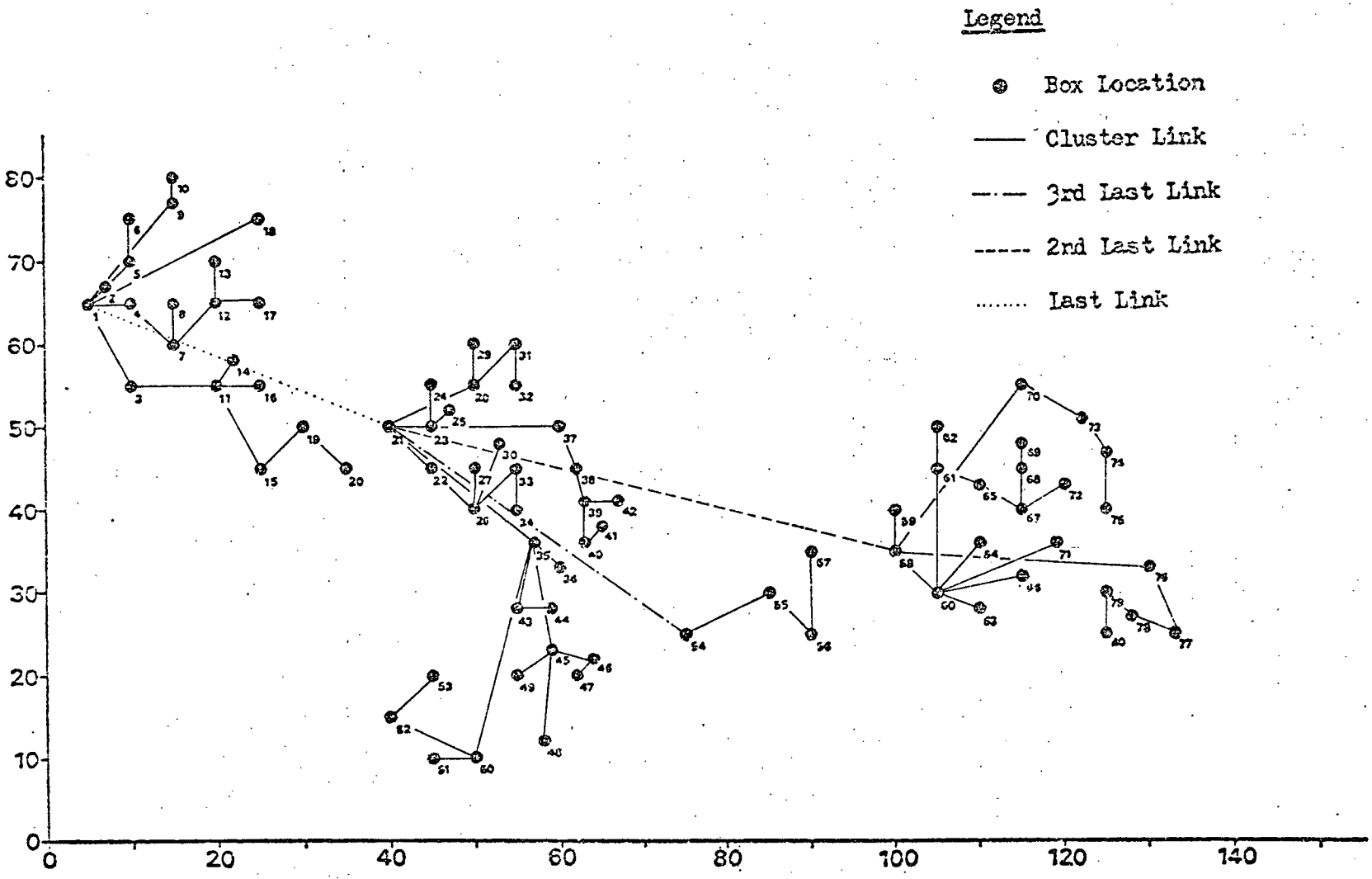


Figure 23. Linkages Outlined by Median Method for DATA2

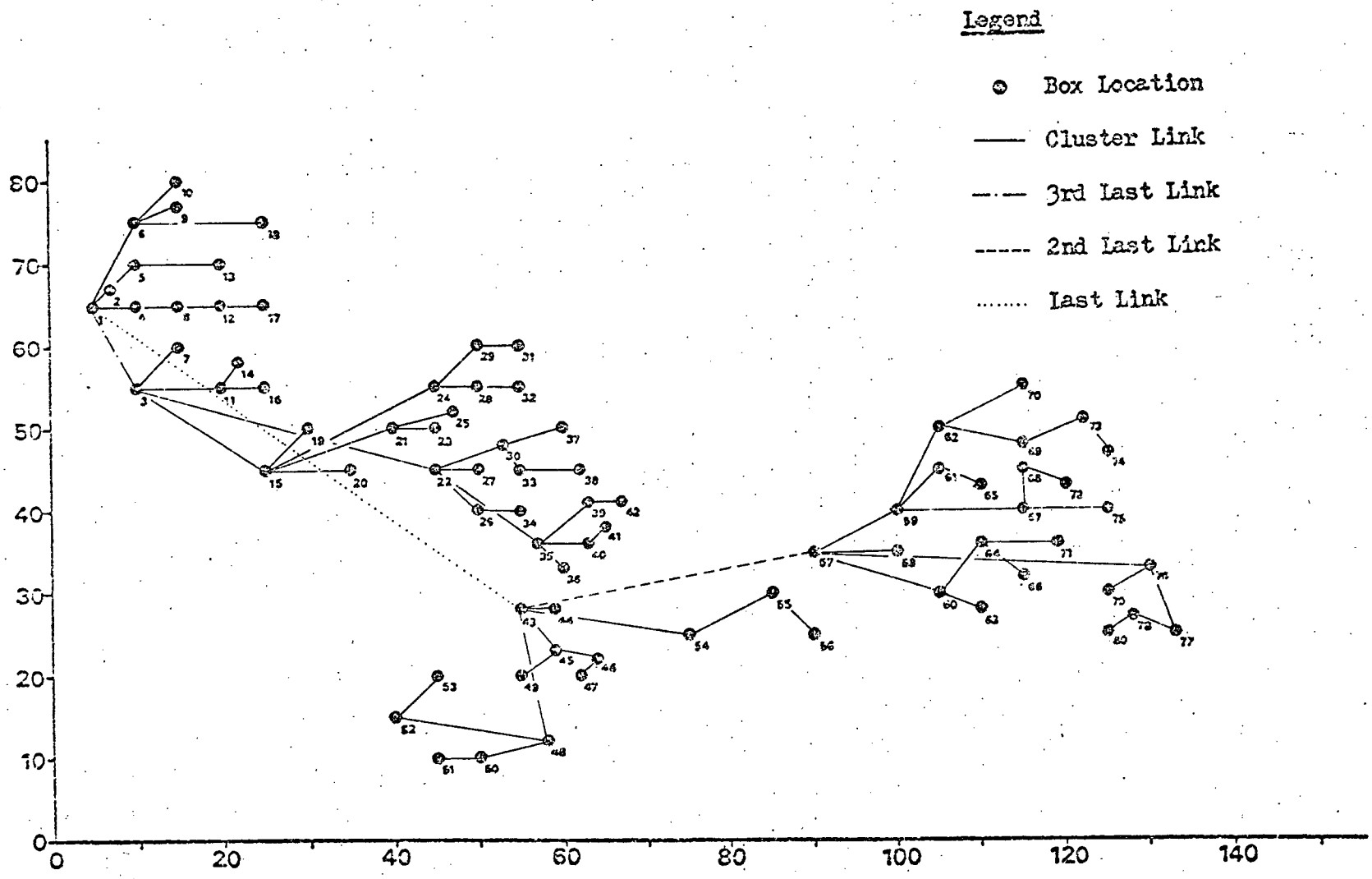


Figure 24. Linkages Outlined by Ward's Method for DATA2

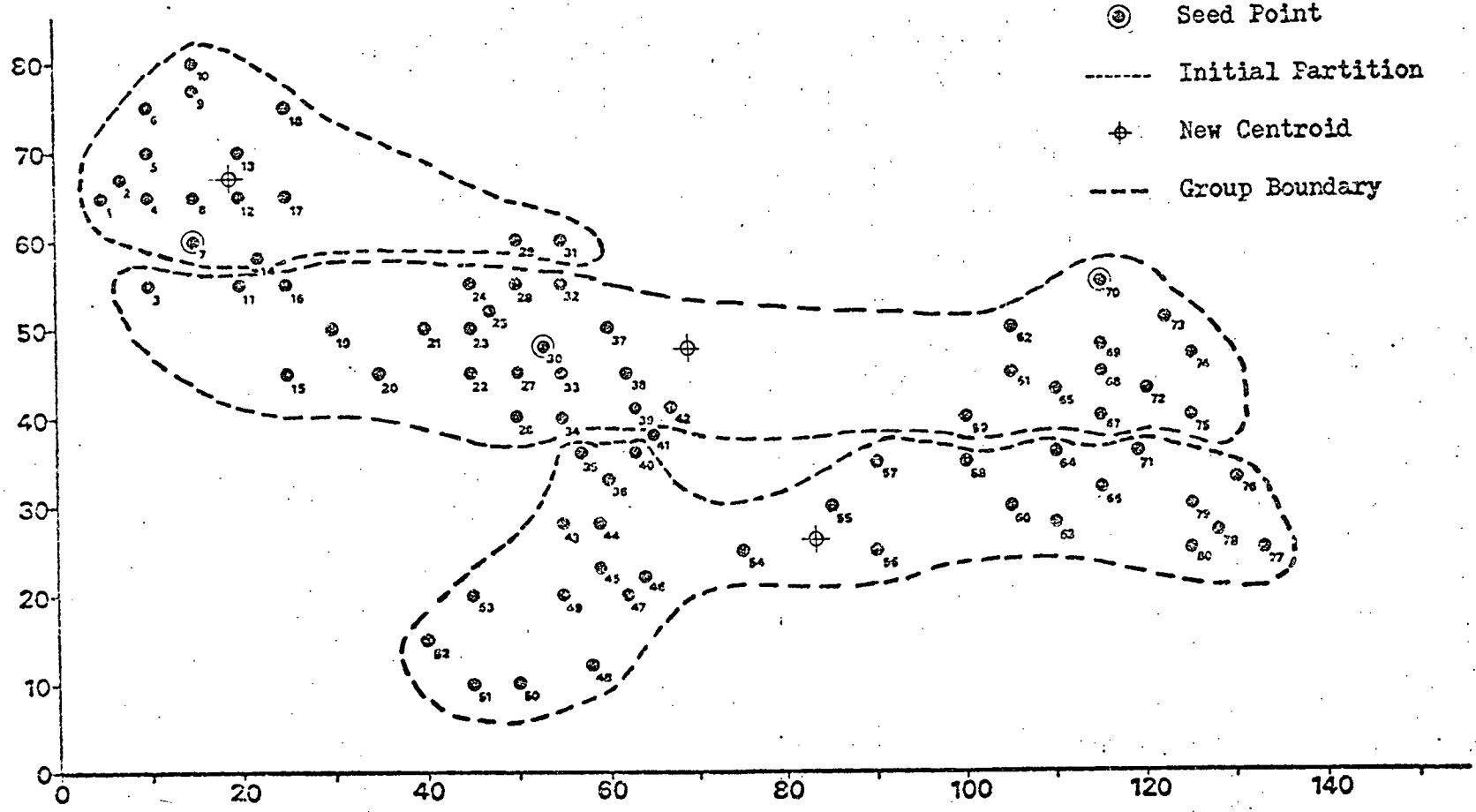


Figure 25. Group Boundaries Defined by Forgy's and Convergent K-mean Methods using Seed Points as Inputs for DATA2

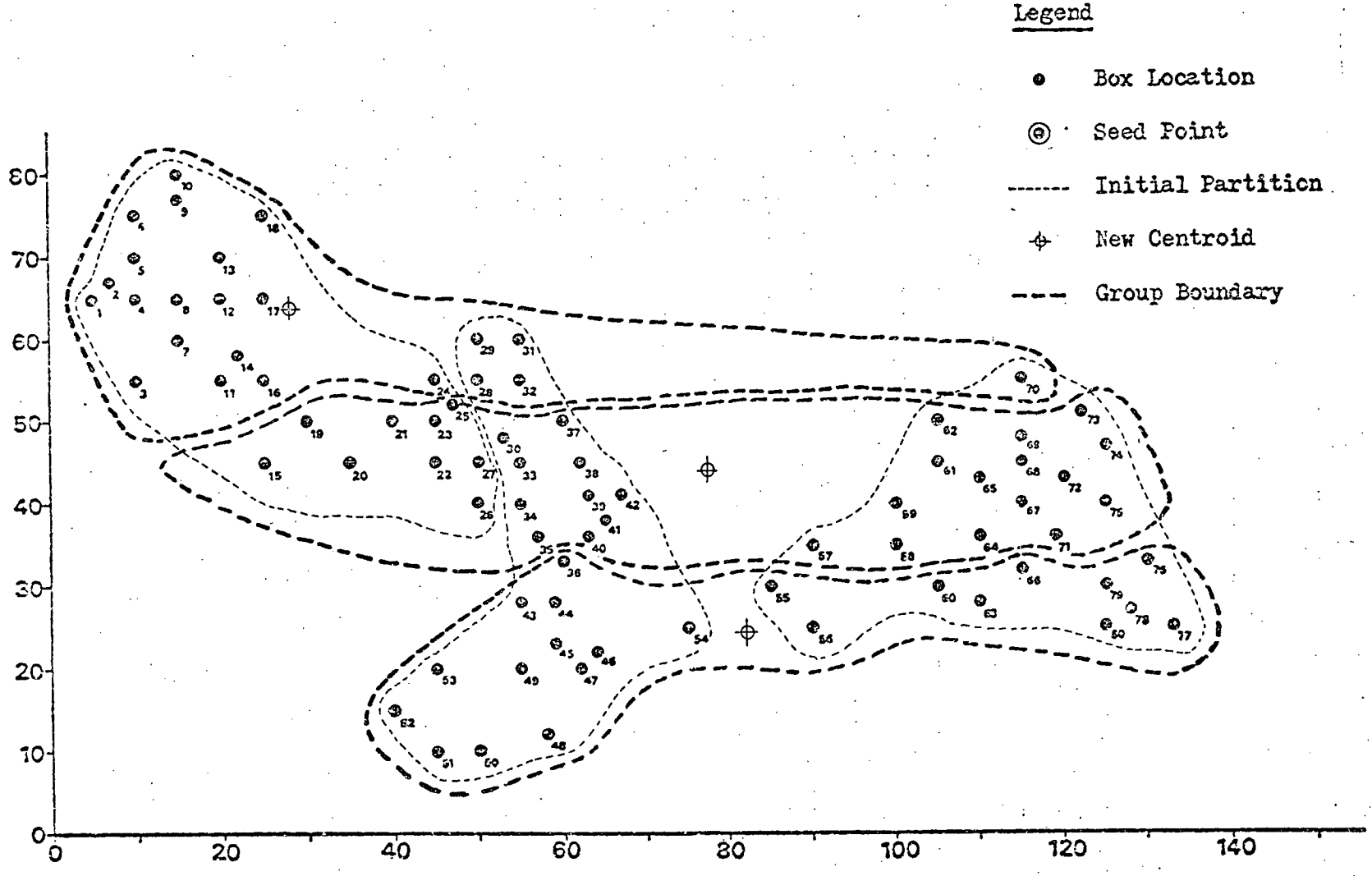


Figure 26. Group Boundaries Defined by Jancey's Method Using Initial Partitions for DATA2

On the whole, it is clear that visually identifiable groups in an unevenly distributed data set can be detected easier than that of evenly distributed data set. Further evaluations are discussed in section 5.5.

5.3.3 North Burnaby Empirical Data (NBDATA)

There are a total of 87 box locations for two routes in the North Burnaby area (Figure 3 and Appendix F). These locations are situated in an area of approximately 34 square miles. This scattered, ungrouped data set has no visually identifiable clusters. These points are presently routed as two groups of 46 and 41 respectively (Figure 27), and these routes has little implication on the desired grouping of the box locations.

The hierarchical and nonhierarchical results are presented in Table 7 and Figures 28-38. The group sizes vary from 3 to 39 for group 1 and 48 to 84 for the other. The results of two dissimilar group sizes in the five hierarchical methods (single linkage:"City - Block" and Euclidean distance; average linkage between merged groups; and the centroid methods) show that a potential cluster (points 1, 2 and 3) located at a distance from the rest of the locations would temper the effectiveness of these clustering algorithms. The only method that groups localities into clusters of 39 and 48 members

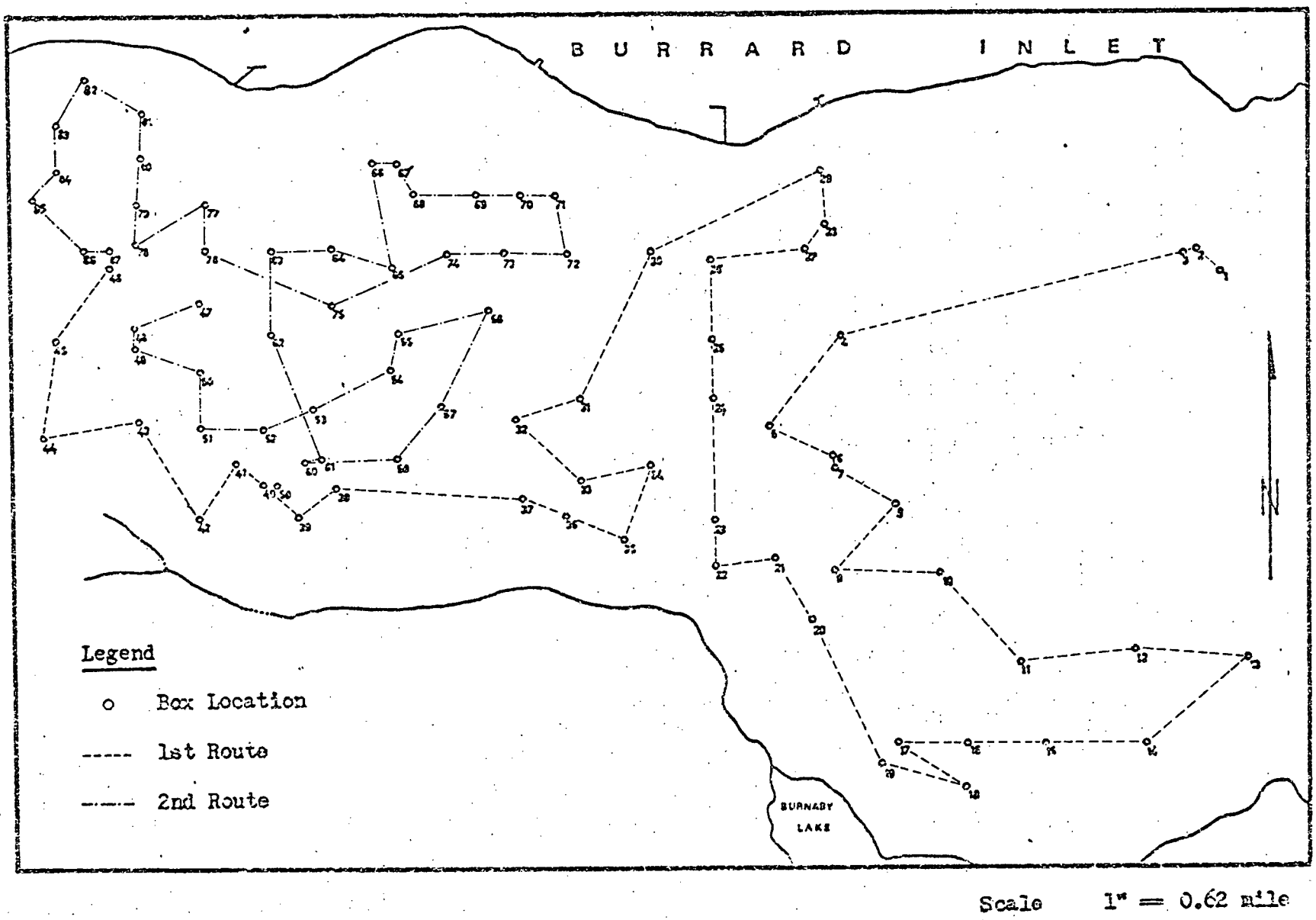


Figure 27. Present Street Letter Box Collection Routes of North Burnaby Area

	Group Sizes	
Group	1	2
Methods		
Hierarchical		
Single Linkage		
City - Block	3	84
Euclidean Distance	3	84
Chi-Squares	29	58
Complete Linkage	37	50
Avg. Linkage between Merged Group	3	84
Avg. Linkage within New Group	39	48
Centroid Method	12	75
Median Method	12	75
Ward's Method	35	52
Nonhierarchical		
Jancey's Method	35	52
Forgy's Method	31	56
Convergent K-mean Method	31	56

Table 7. Results of 12 Clustering Method for NBDATA

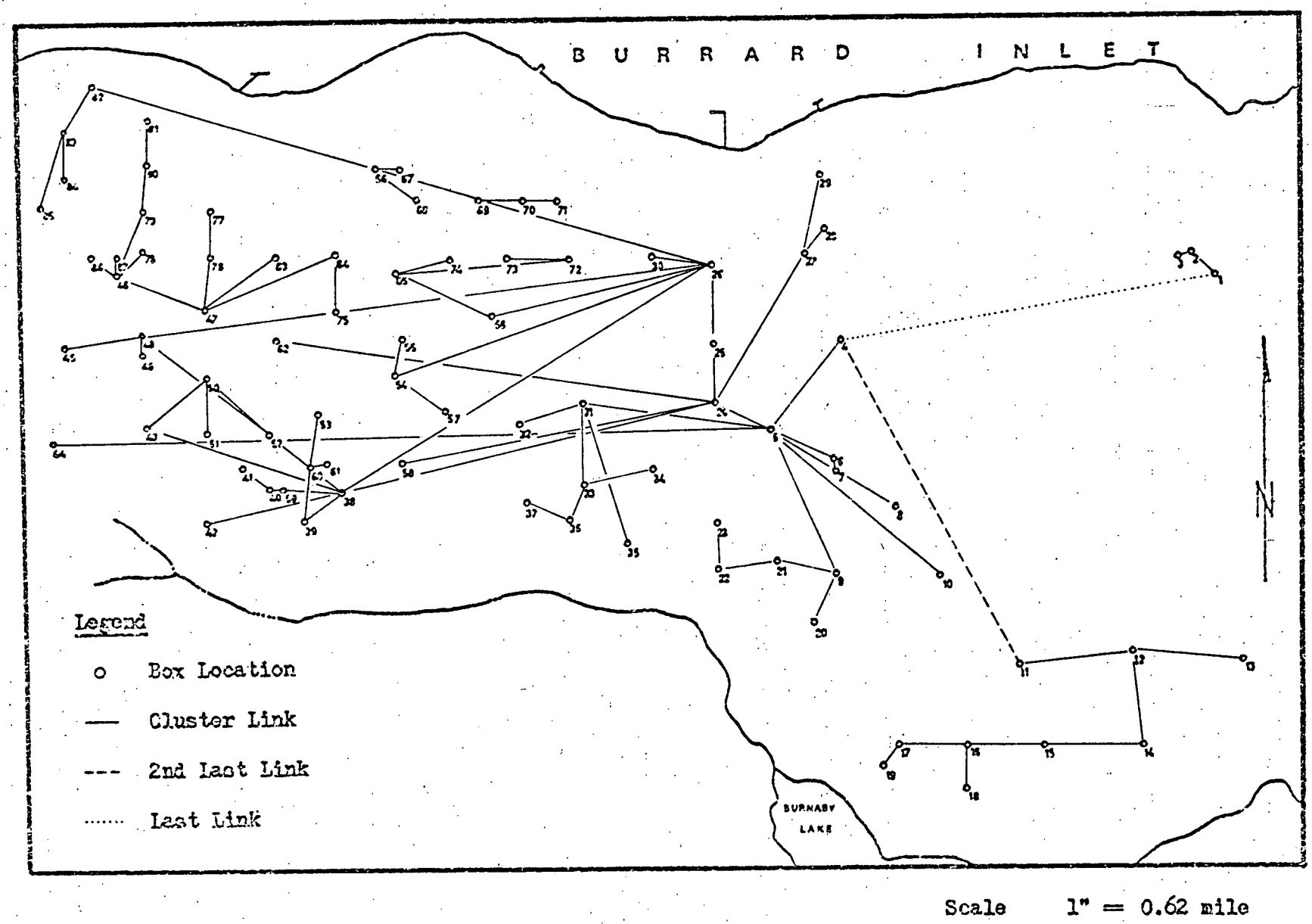


Figure 28. Linkages Outlined by Single Linkage- " City - Block " Method for NBDATA

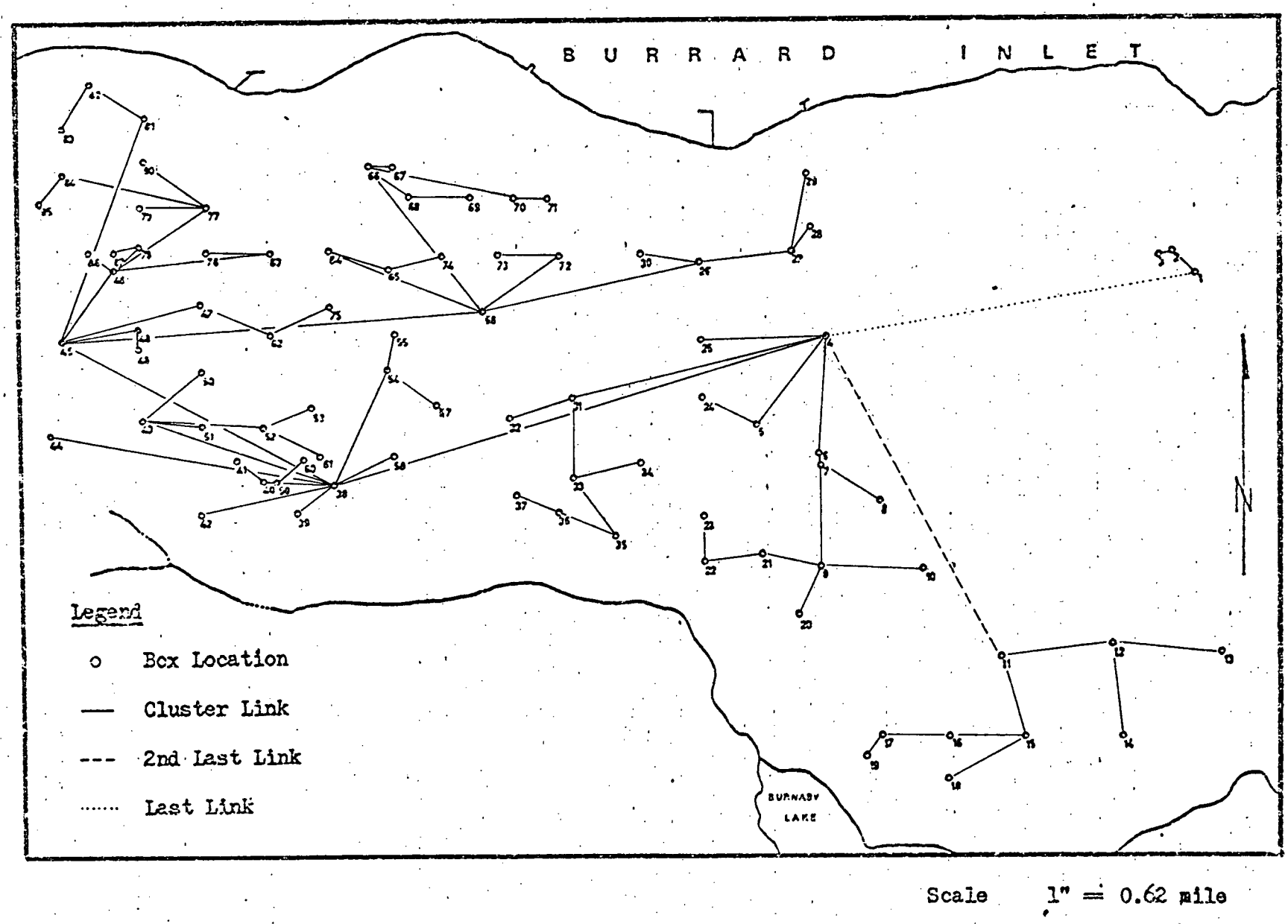


Figure 29. Linkages Outlined by Single Linkage-Euclidean Distance Method for NBDATA

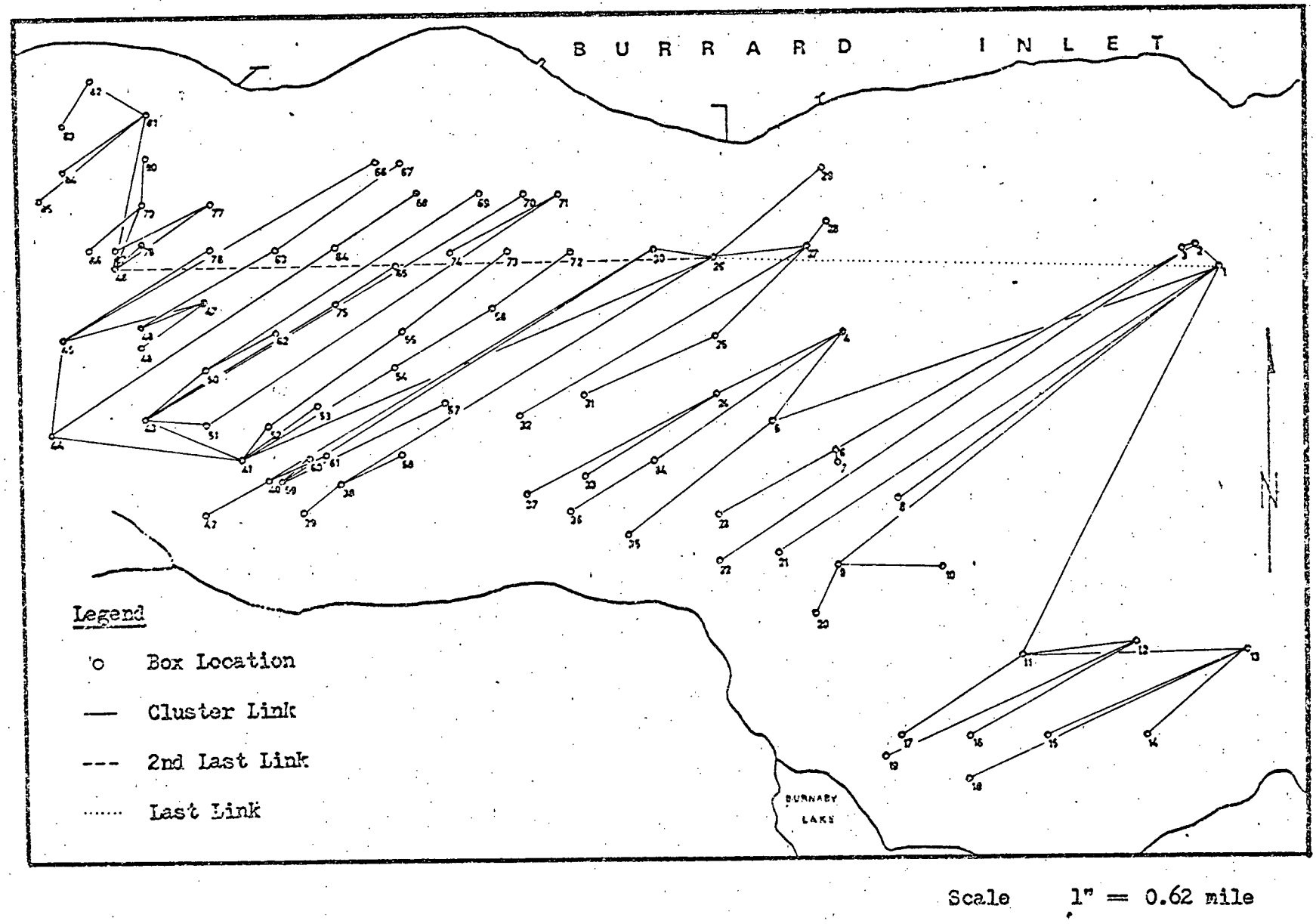


Figure 30. Linkages Outlined by Single Linkage-Chi Squares Method for NBDATA

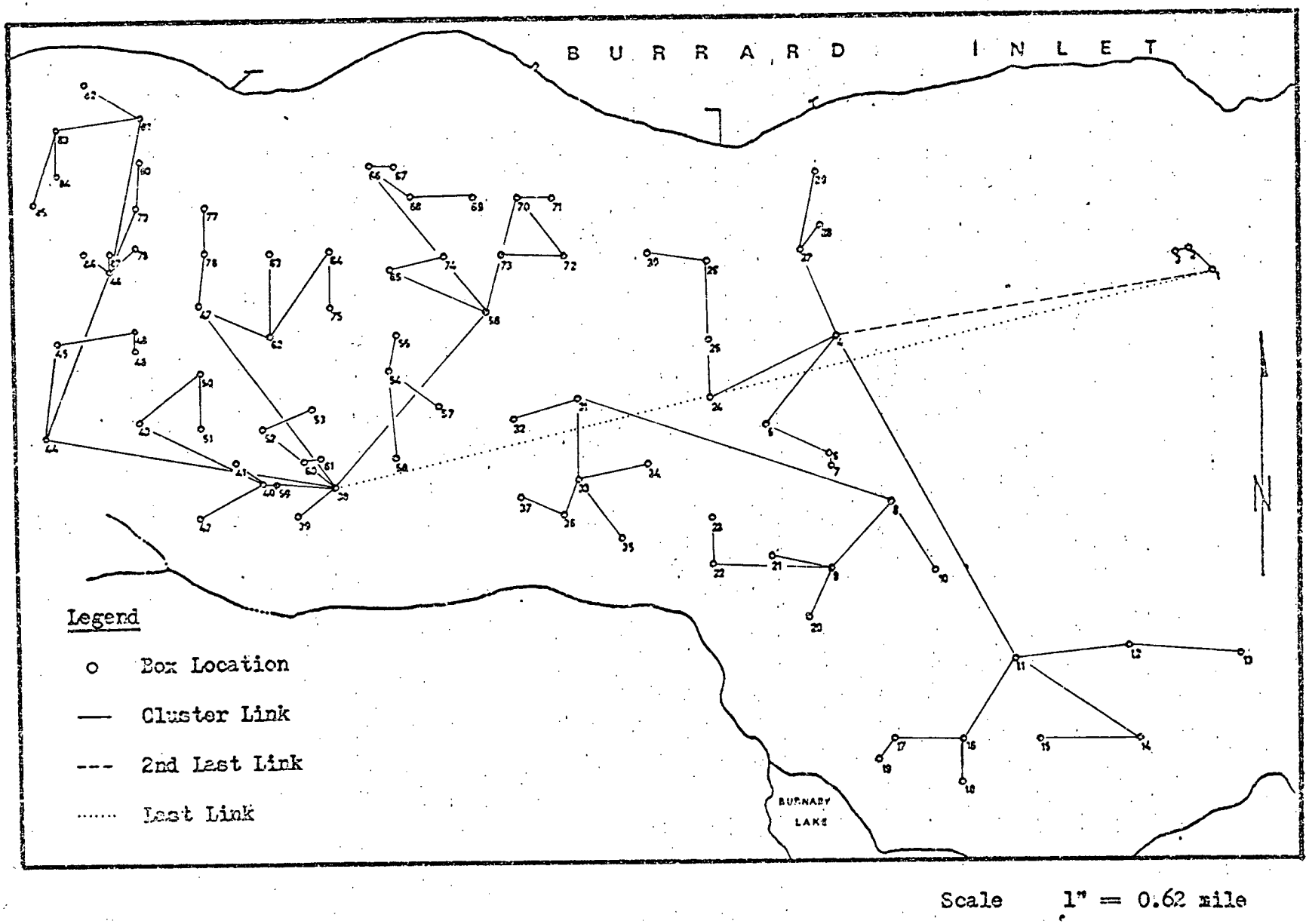


Figure 31. Linkages Outlined by Complete Linkage Method for NBDATA

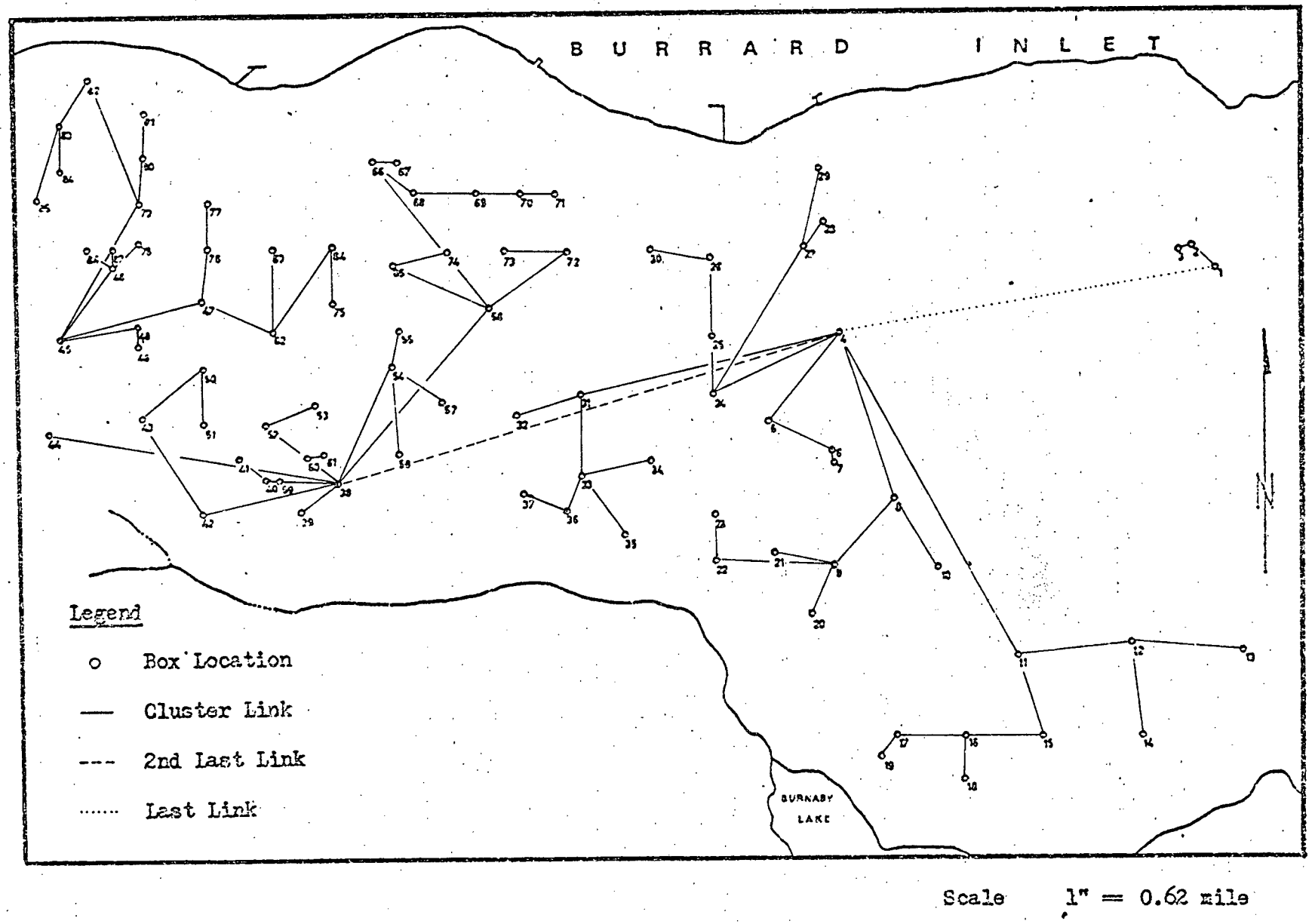


Figure 32. Linkages Outlined by Avg. Linkage between Merged Groups Method for NBDATA

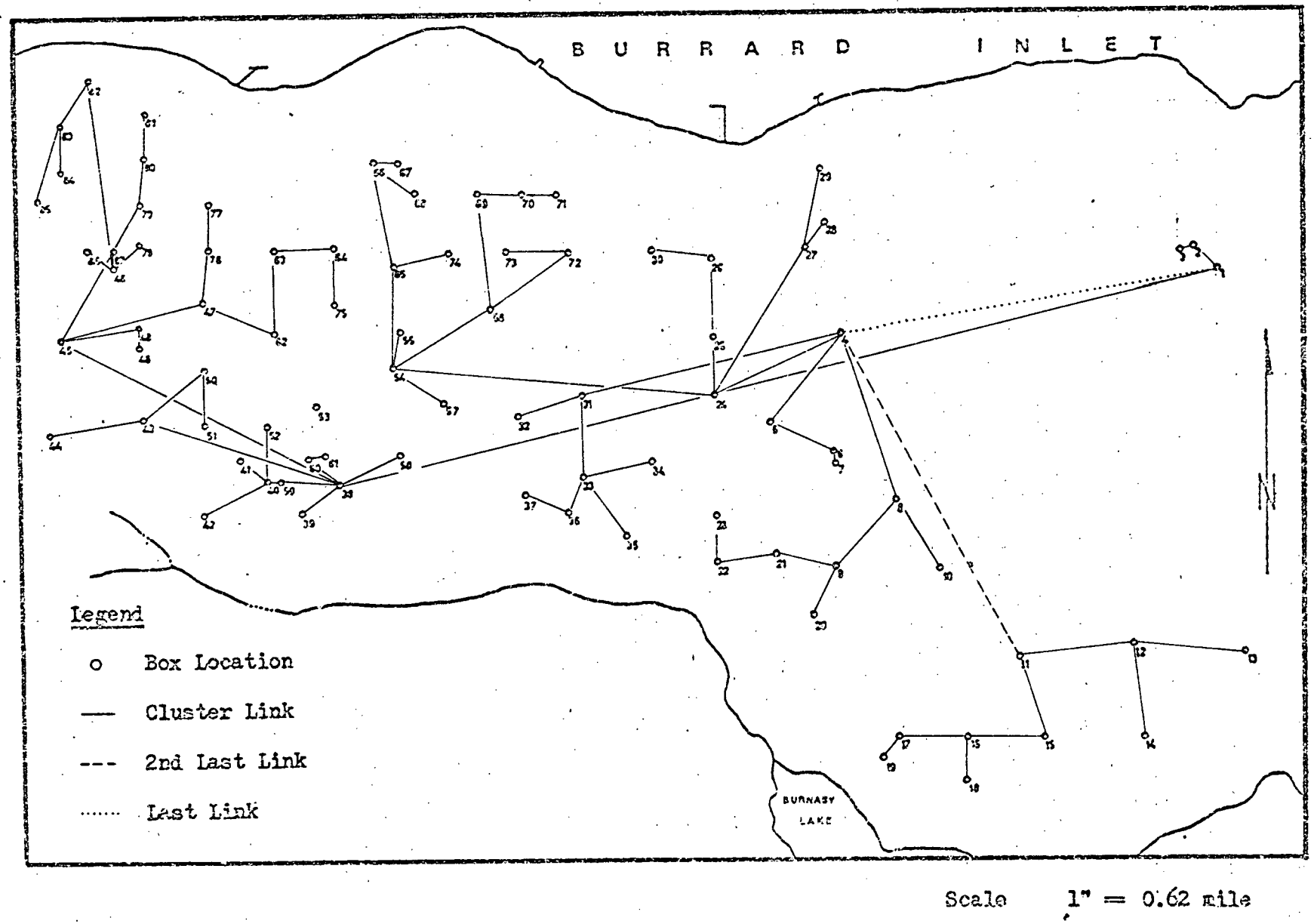


Figure 33. Linkage Outlined by Avg. Linkage within New Group Method for NBDATA

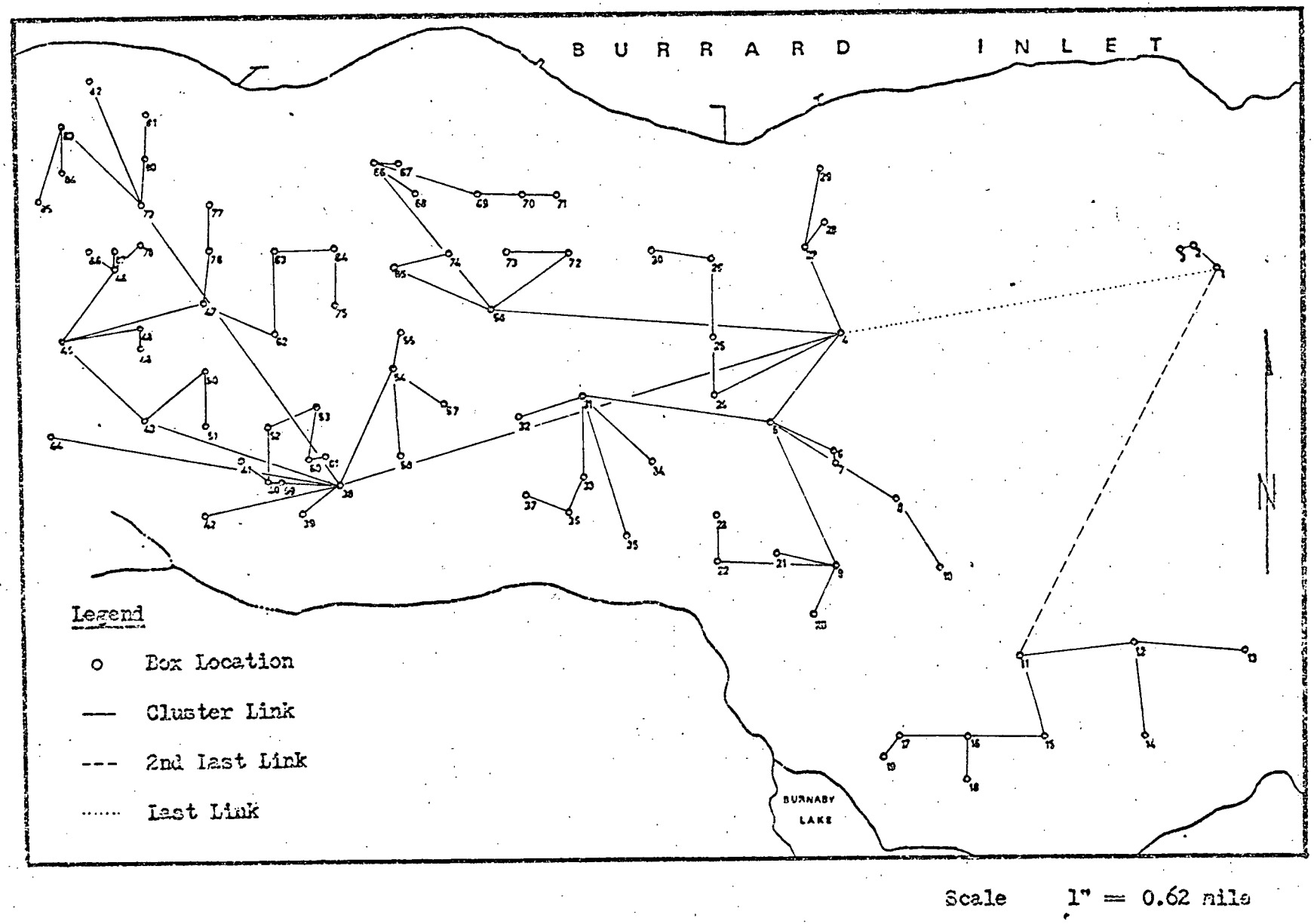


Figure 34. Linkages Outlined by Centroid Method for NBDATA

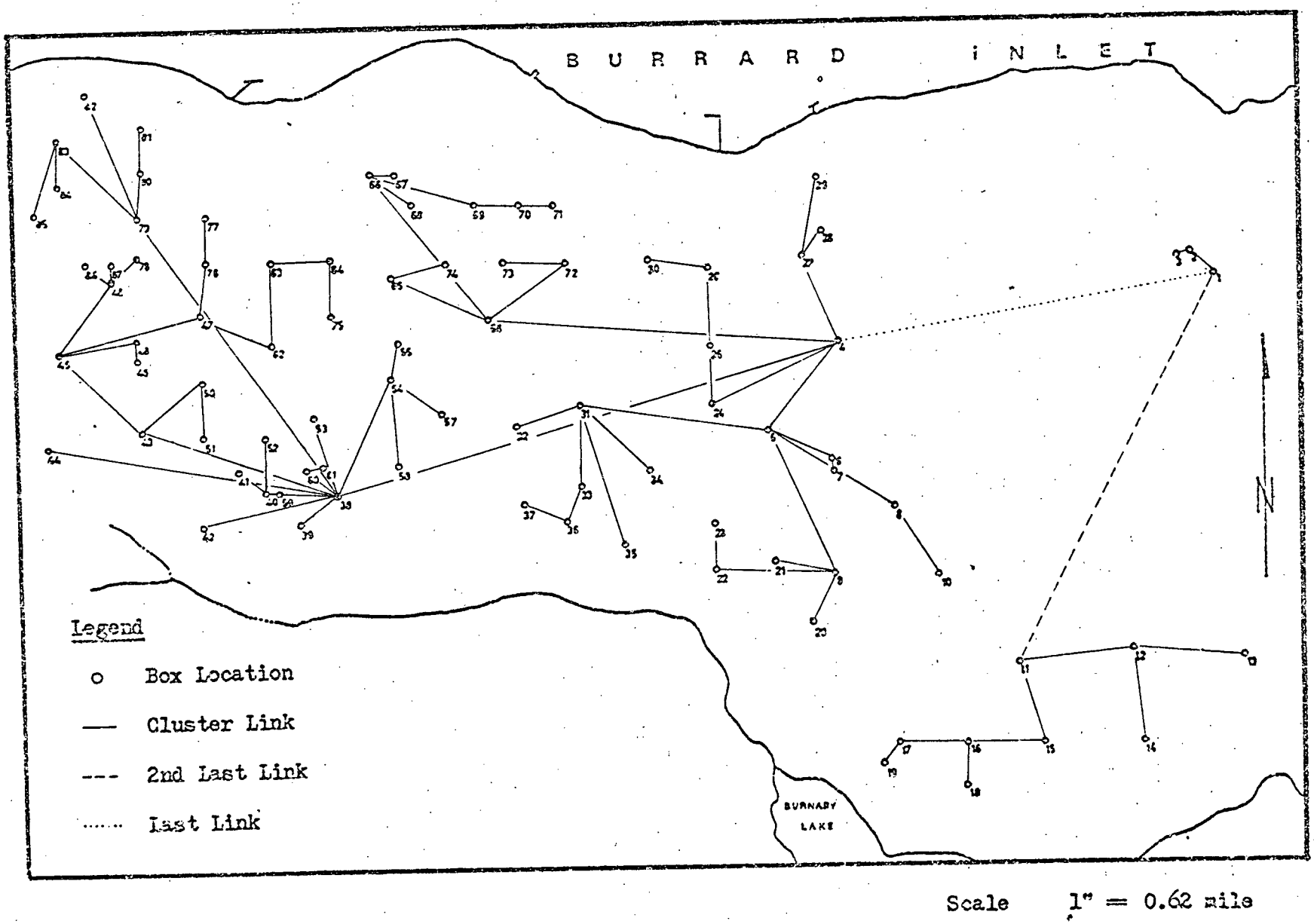


Figure 35. Linkages Outlined by Median Method for NBDATA

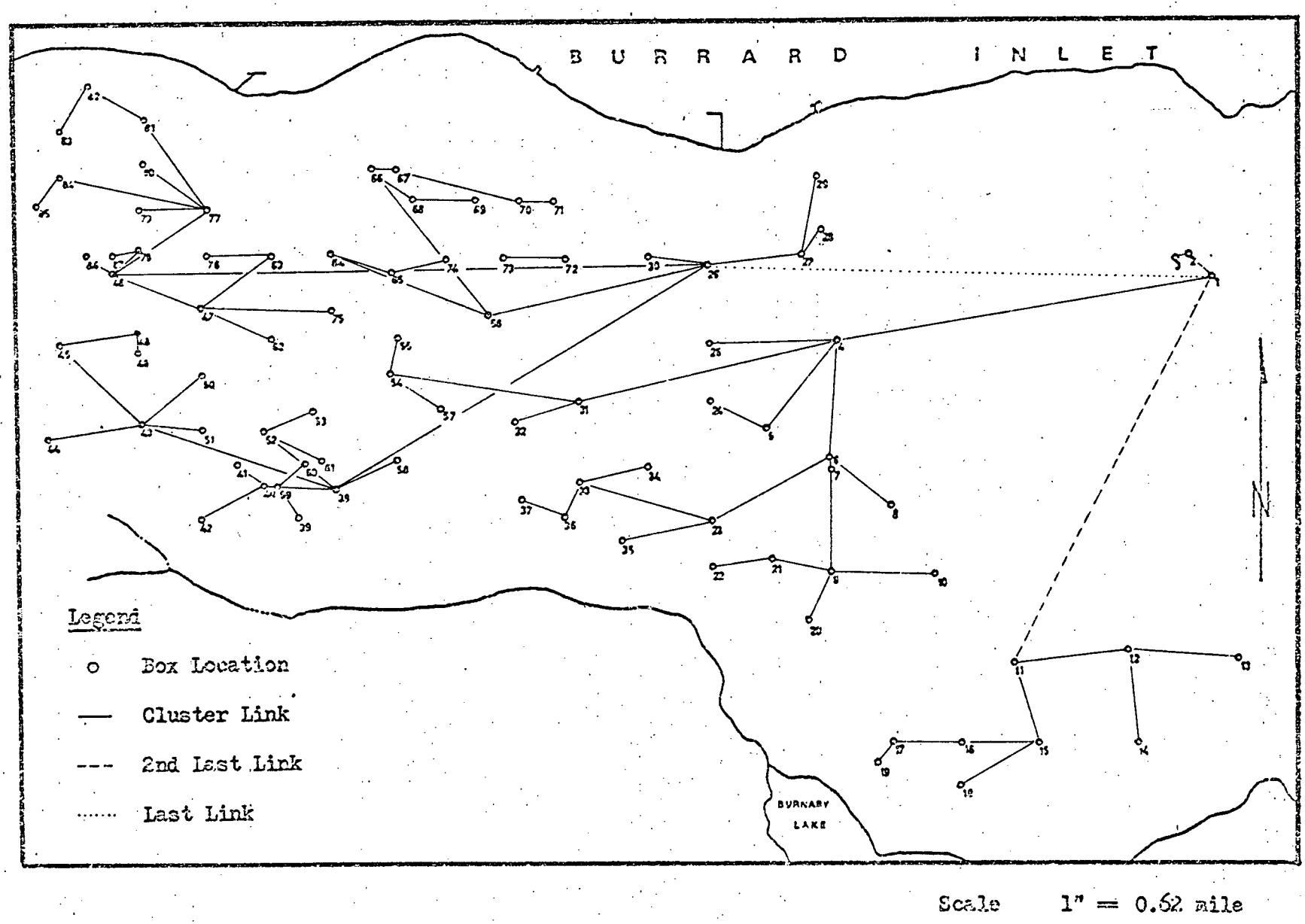


Figure 36. Linkages Outlined by Ward's Method for NBDATA

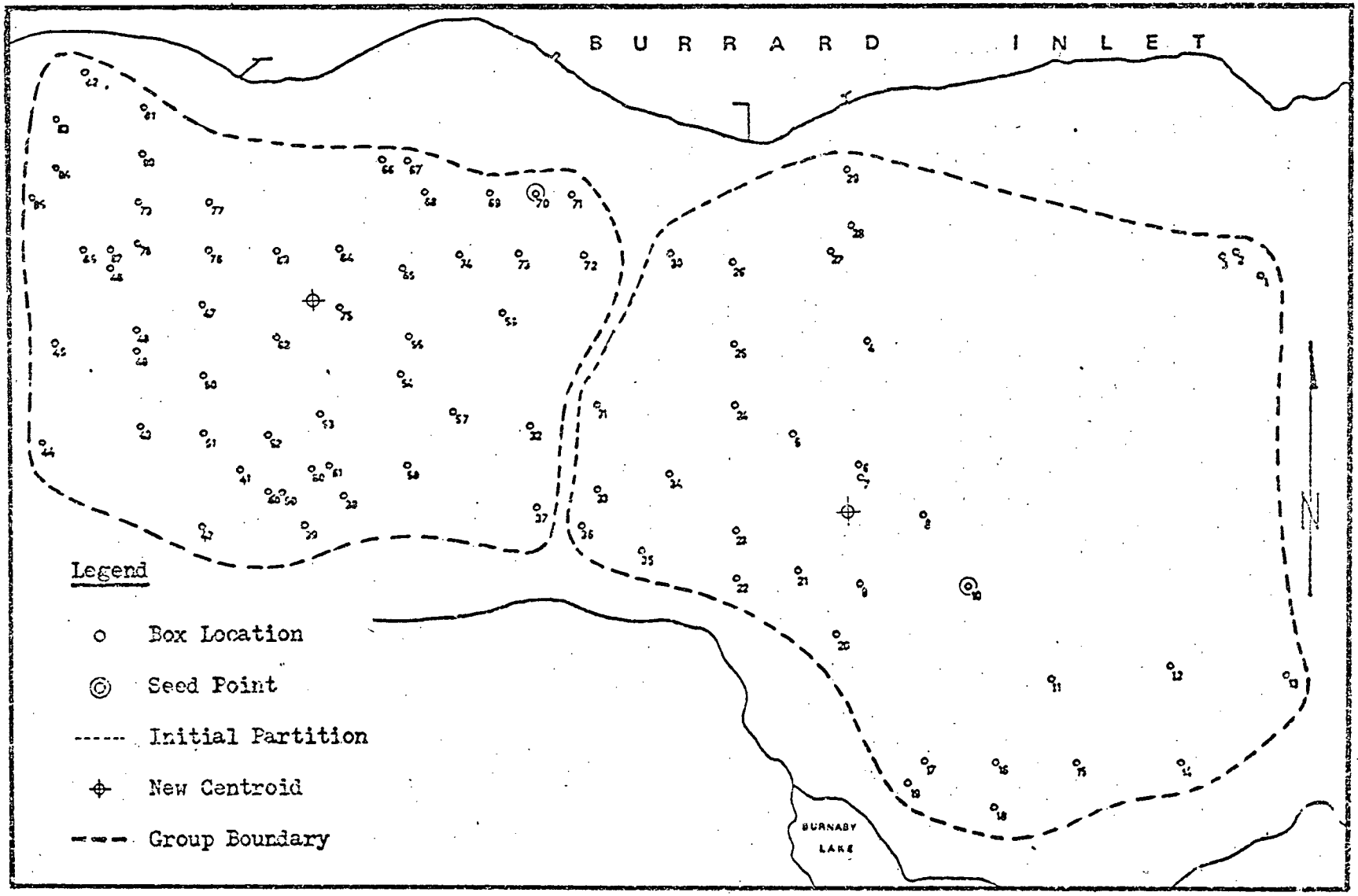


Figure 37. Group Boundaries Defined by Jancey's Method Using Seed Points as Inputs for NBDATA

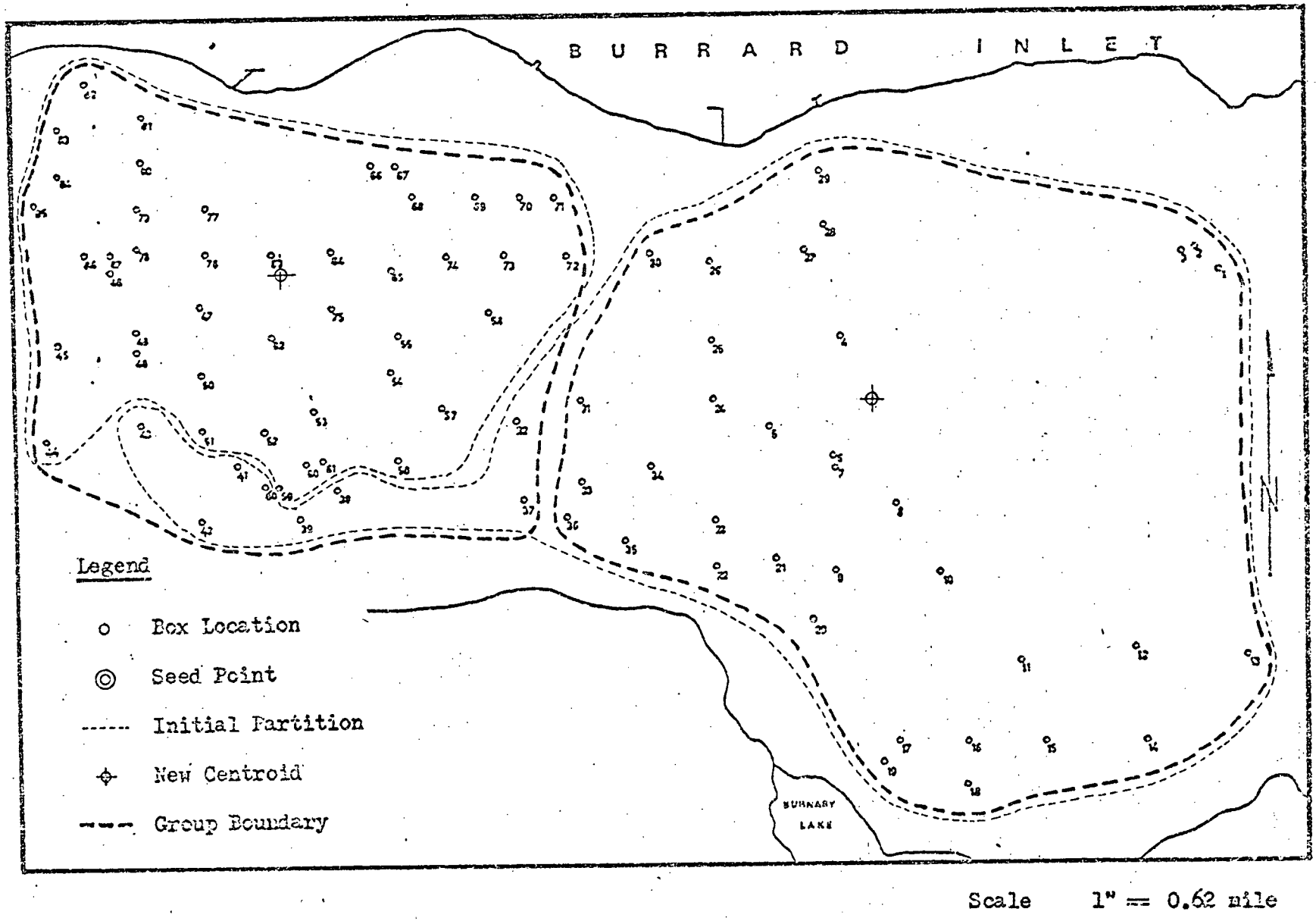


Figure 38. Group Boundaries Defined by Forgy's and Convergent K-mean Methods Using Initial Partition for NBDATA

respectively is the average linkage within the new group algorithm. This, however, does not indicate that this method is most suitable for this set of data. The evaluation approach in latter sections would give an in-depth examination of these results.

Three sets of seed points and initial partitions were also used to test the effect in initial cluster on the results (Table 8). There are indications that initial partitions have less influence than seed points on the data set. These variances in resulting group sizes, however, do not indicate the superiority of one nonhierarchical method over the others.

5.3.4 South Burnaby Empirical Data (SBDATA)

This set of data depicts the locations of 113 mail boxes which are serviced by 3 truck routes (Figure 39) in the South Burnaby area. There is no distinct group boundaries that are visually identifiable (Figure 4 and Appendix F) for this set. This probably represents the typical distribution of mail boxes in other areas. The 12 clustering methods used previously for the other three sets are also employed for grouping this data set.

	Group Sizes					
Trial	1		2		3	
Group	1	2	1	2	1	2
Seed Points Methods	5	62	10	70	15	55
Jancey's	34	53	35	52	34	53
Forgy's	35	52	31	56	31	56
Convergent K-mean	35	52	31	56	31	56
Initial Partition Methods	40	47	43	44	35	52
Jancey's	35	52	34	53	35	52
Forgy's	35	52	35	52	35	52
Convergent K-mean	35	52	35	52	35	52

Table 8. Summary of Nonhierarchical Runs for NBDATA

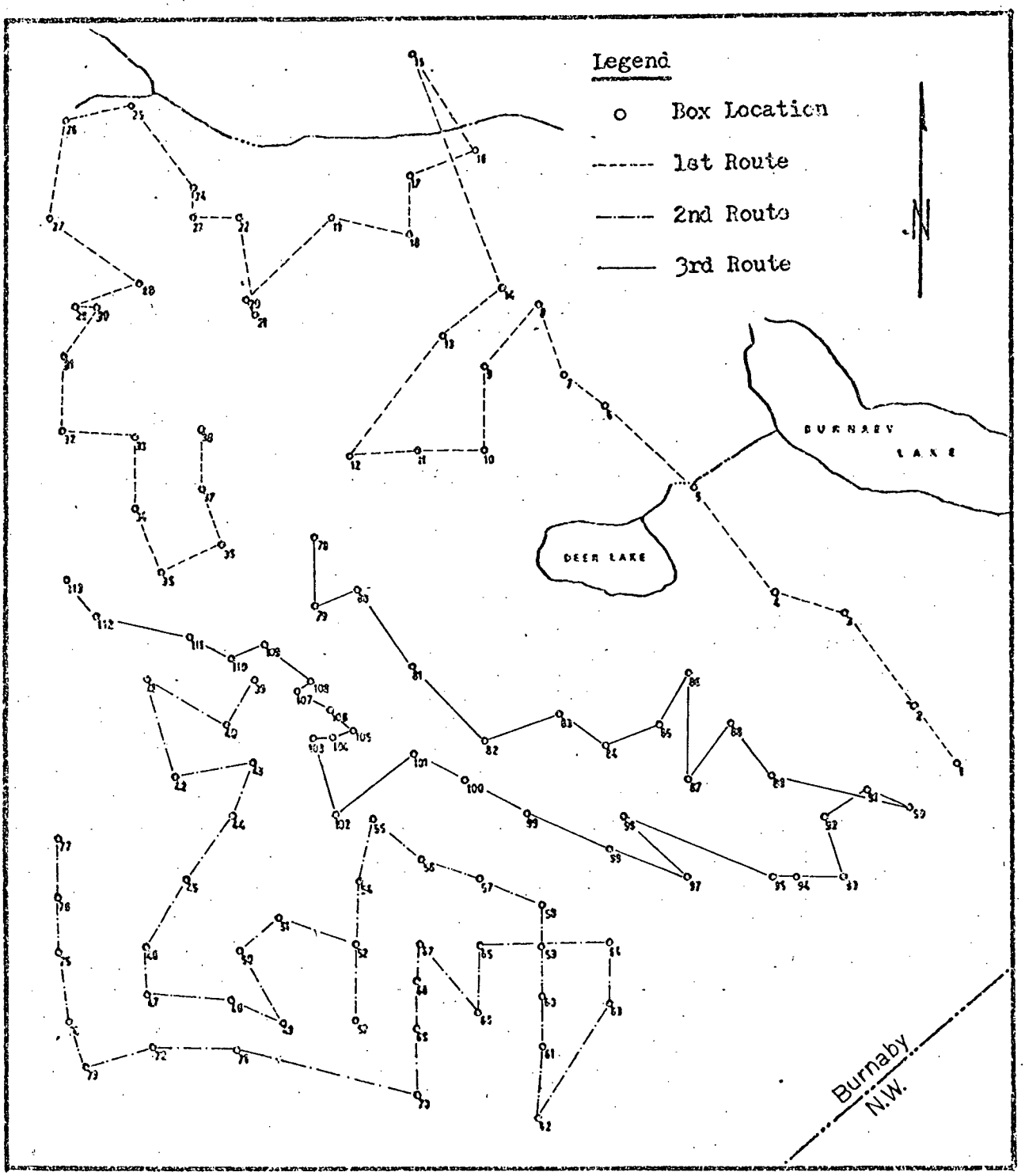


Figure 39. Present Street Letter Box Collection Routes of South Burnaby Area

The results of these 12 methods are tabulated and plotted in Table 9 and Figures 40-50. The single linkage-" City - Block " method identifies group sizes of 1, 1 and 111. These group sizes are totally unacceptable as aids to scheduling problems. The centroid methods have similar unbalanced results. The complete linkage, the Ward's method and the three nonhierarchical algorithms give reasonably balanced number of members in each group. However, all the resultant group sizes from these clustering methods do not resemble the actual number of call points (group sizes 38, 40 and 36). A more extensive evaluation examining the critical group sizes will be discussed in Section 5.5.

The nonhierarchical methods' results presented in Table 9 are the more distinctive group sizes of six trials using different seed points and partitions (Table 10). These trials show little significant differences in the group sizes.

5.4 Tools for Evaluation

Two methods were developed to evaluate the groups resulting from various techniques on the four sets of data. Both methods are included in the computer program ROUTE (Appendix H) doing statistics and routing on the groups outlined by the different clustering methods.

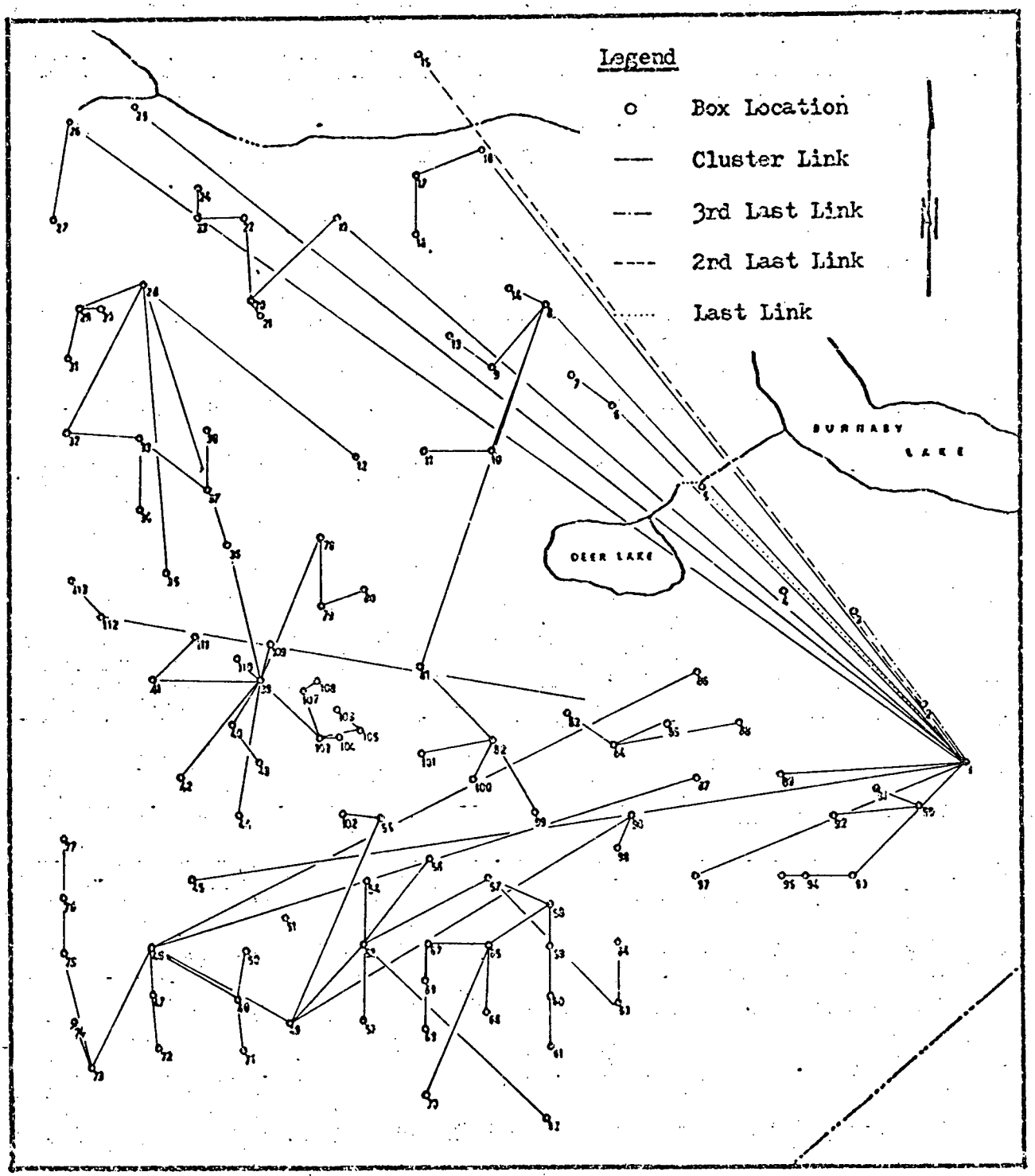
	Group Sizes		
Group	1	2	3
Methods			
Hierarchical			
Single Linkage			
City - Block	1	1	111
Euclidean Distance	21	26	66
Chi-Squares	34	36	43
Complete Linkage	20	45	48
Avg. Linkage between Merged Group	21	26	66
Avg. Linkage within New Group	22	36	55
Centroid Method	2	18	93
Median Method	2	18	93
Ward's Method	45	28	40
Nonhierarchical			
Jancey's Method	34	45	34
Forgy's Method	33	46	34
Convergent K-mean Method	33	46	34

Table 9. Results of 12 Clustering Methods for SBDATA

Group Sizes

Trial	1			2			3		
Group	1	2	3	1	2	3	1	2	3
Methods \ Seed Points	20	44	87	15	40	93	20	39	92
Jancey's	35	44	34	34	45	34	34	45	34
Forgy's	33	46	34	33	46	34	34	49	30
Convergent K-mean	33	46	34	34	49	30	34	49	30
Methods \ Initial Partition	35	40	38	38	38	37	35	44	34
Jancey's	34	45	34	35	44	34	33	46	34
Forgy's	36	46	31	36	47	30	36	46	31
Convergent K-mean	35	44	34	35	44	34	35	44	34

Table 10. Summary of Nonhierarchical Runs for SBDATA



Scale 1" = 0.62 mile

Figure 40. Linkages Outlined by Single Linkage-
" City - Block " Method for SBDATA

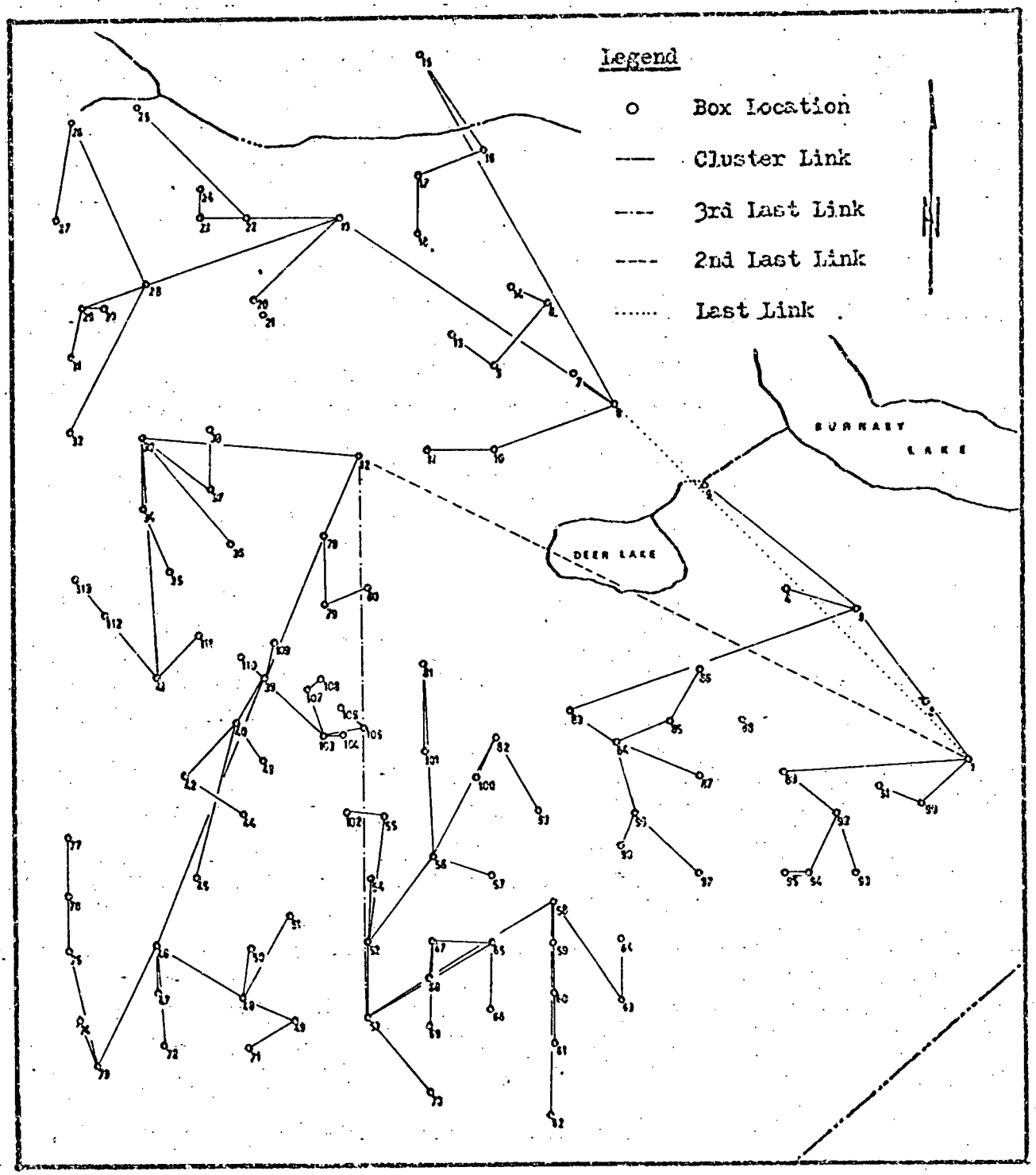


Figure 41. Linkages Outlined by Single Linkage-Euclidean Distance Method for SBDATA

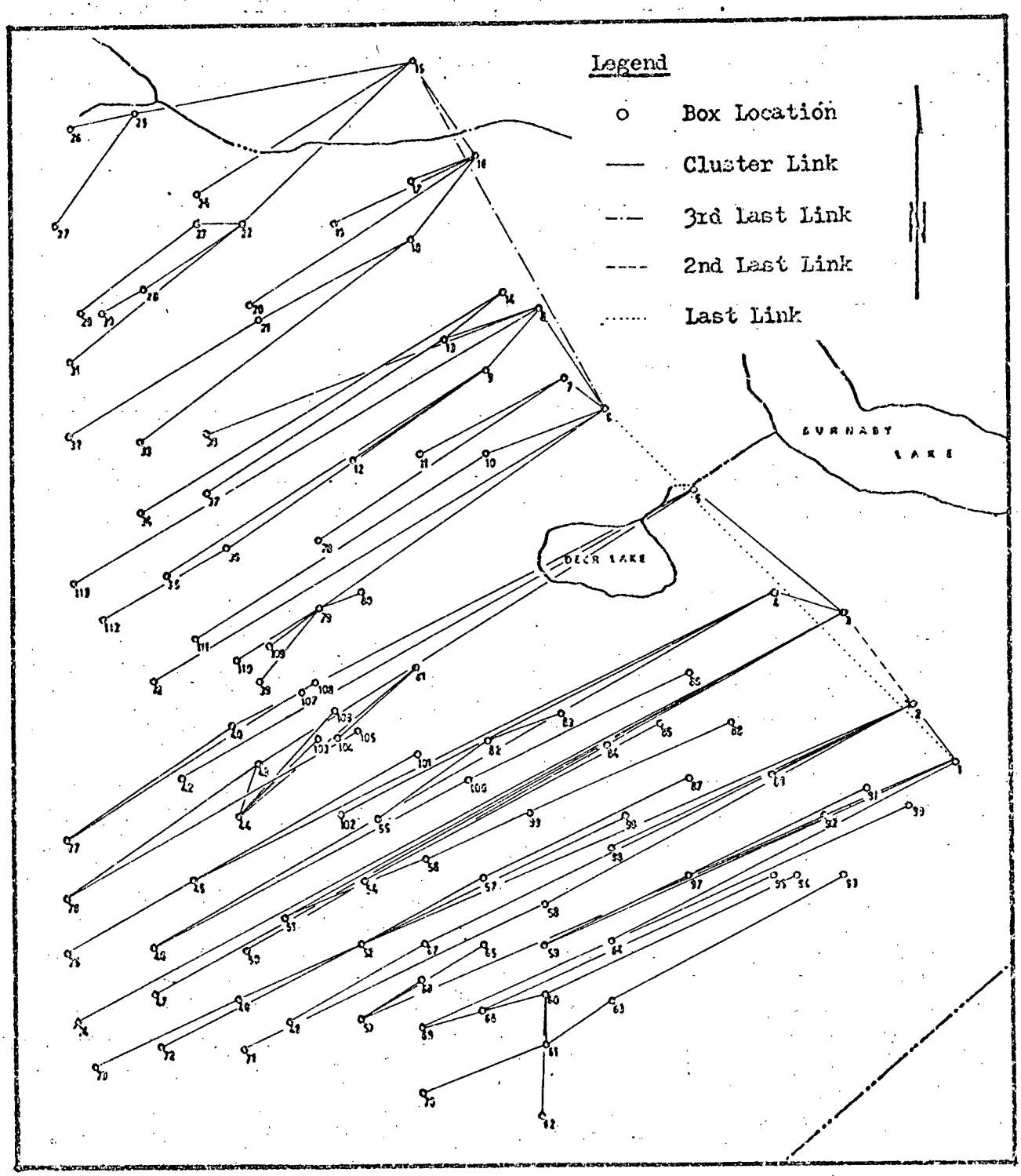


Figure 42. Linkages Outlined by Single Linkage-Chi-Squares Method for SBDATA

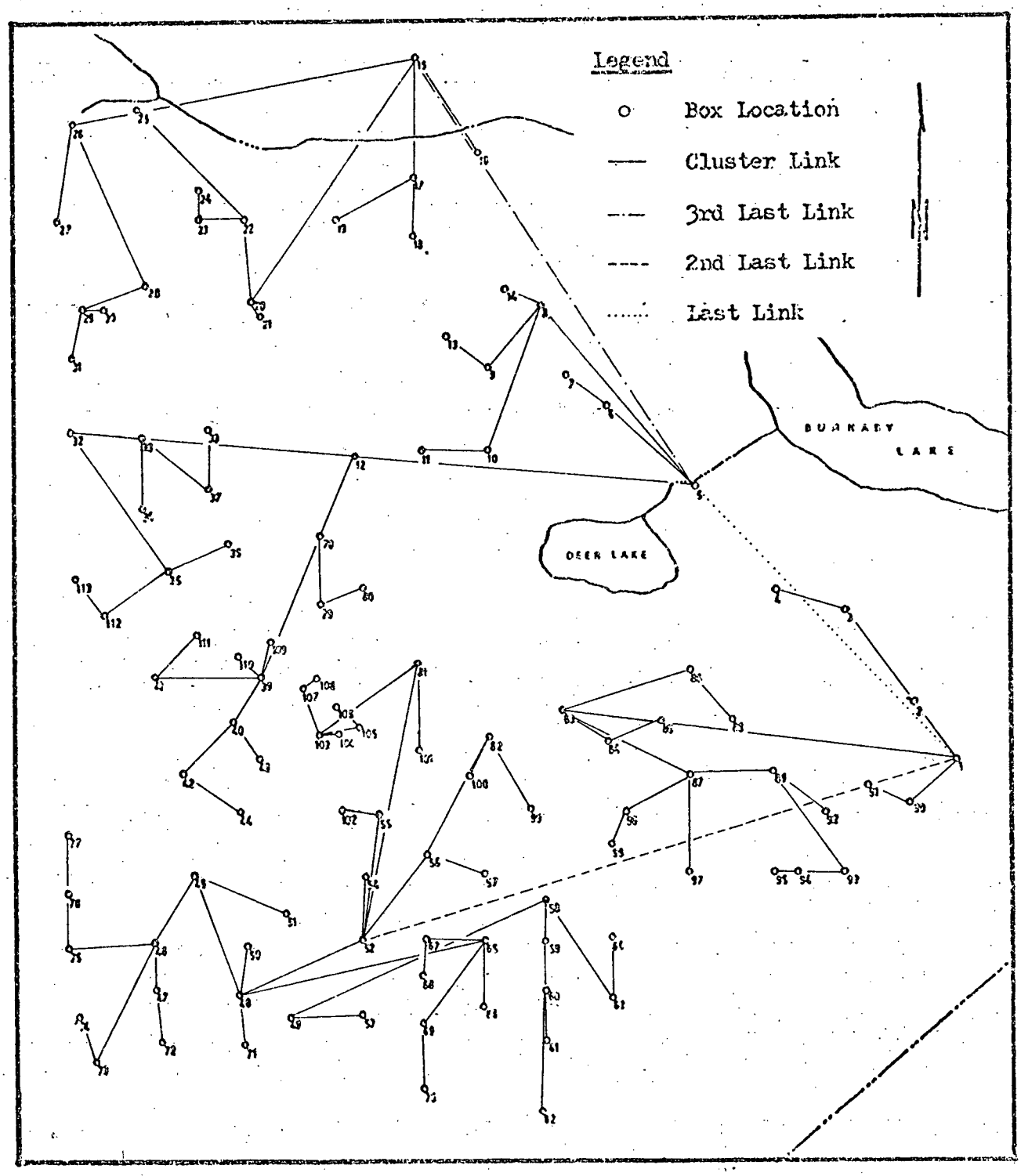


Figure 43. Linkages Outlined by Complete Linkage Method for SBDATA

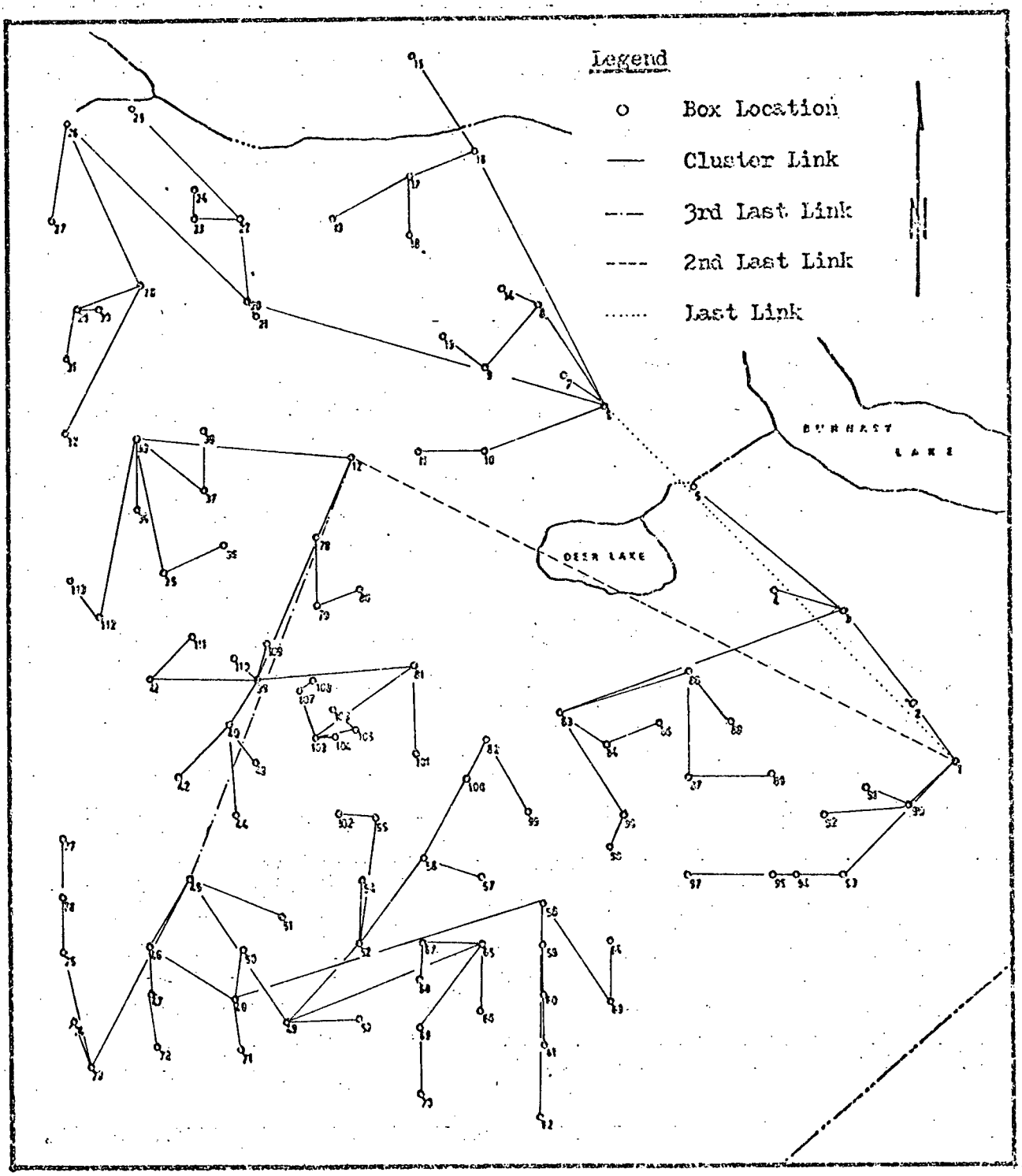


Figure 44. Linkages Outlined by Avg. Linkage between Merged Groups Method for SBDATA

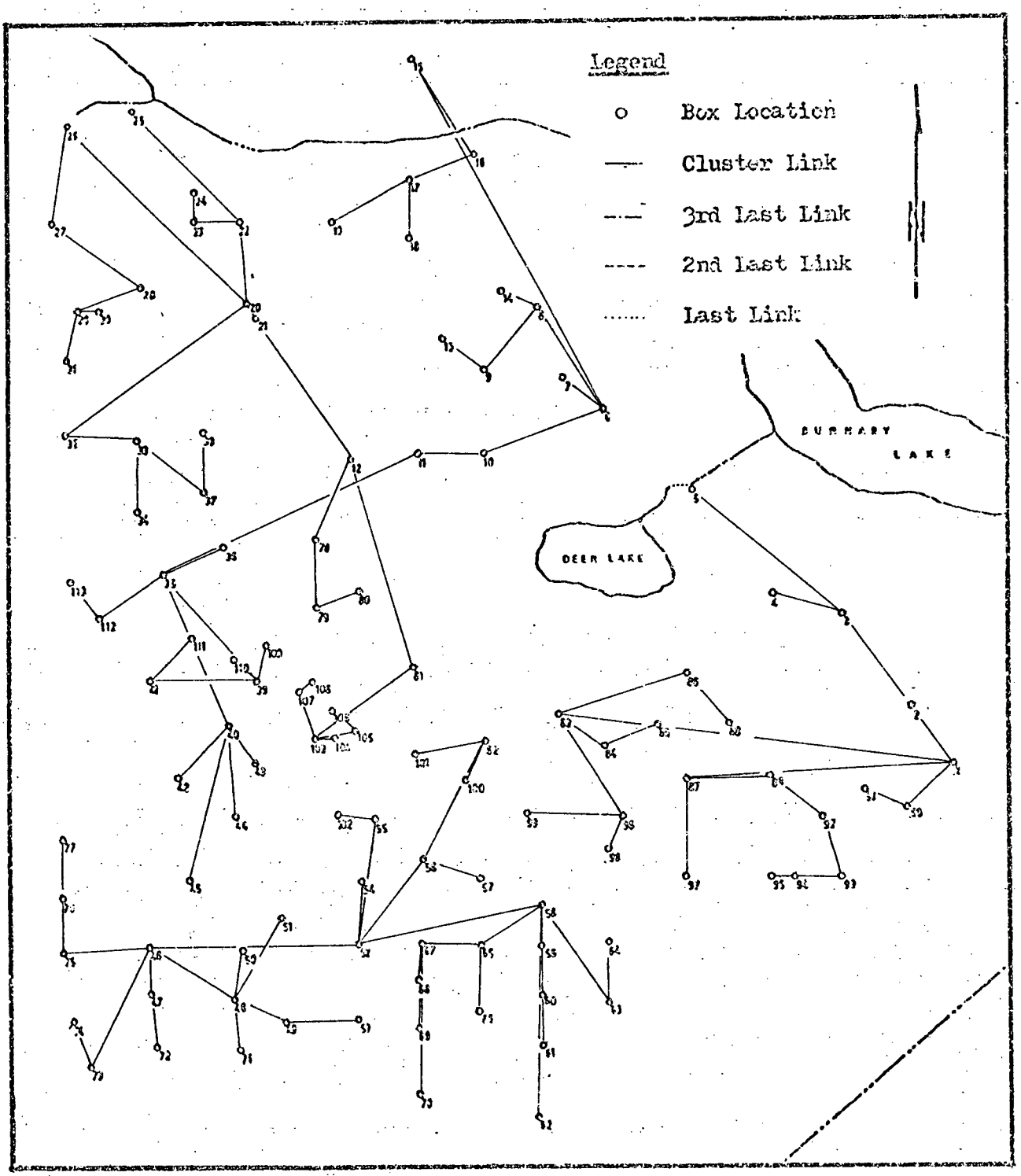


Figure 45. Linkages Outlined by Avg. Linkage within New Group Method for SBDATA

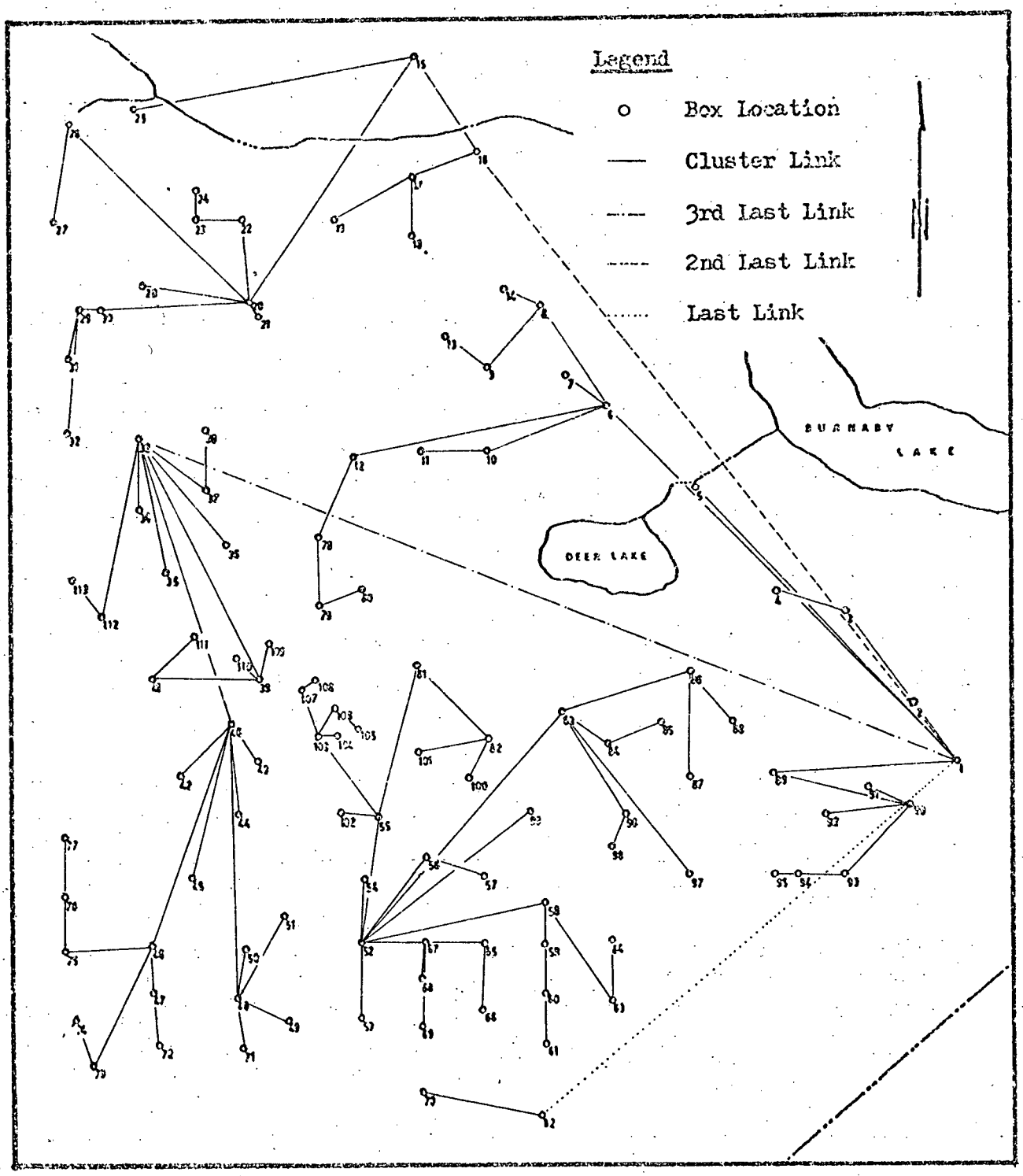


Figure 46. Linkages Outlined by Centroid Mehod for SBDATA

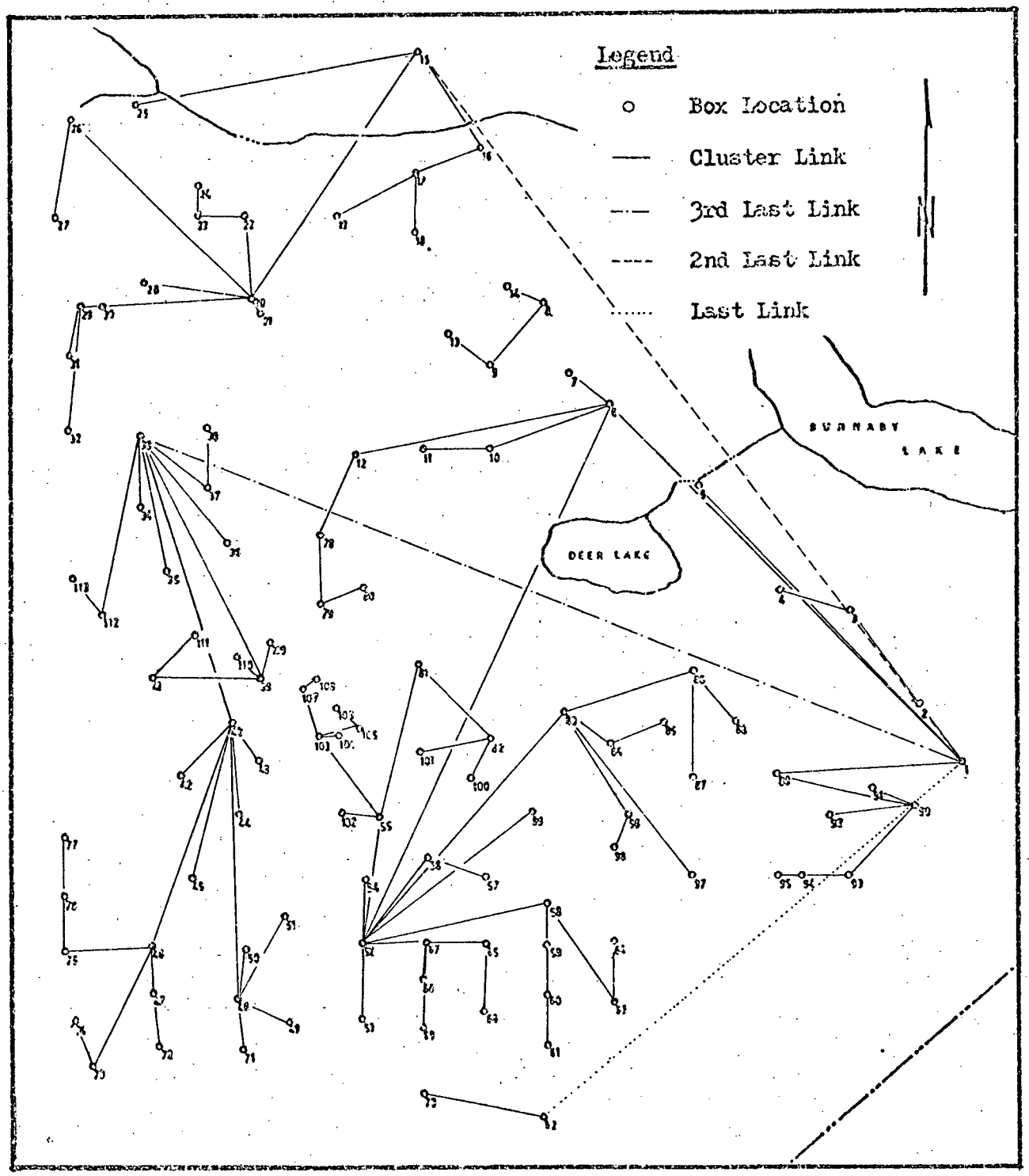


Figure 47. Linkages Outlined by Median Method for SBDATA

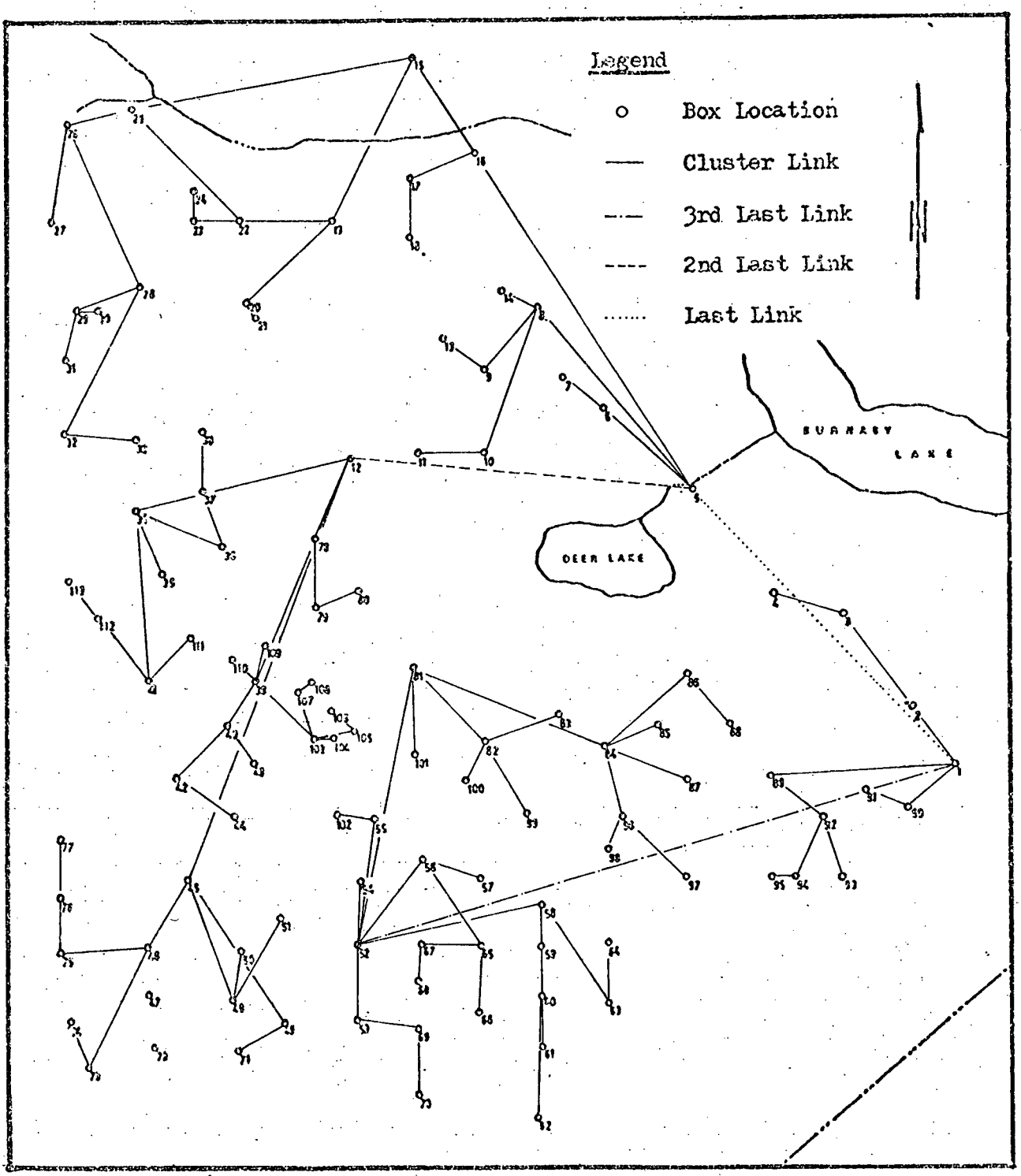


Figure 48. Linkages Outlined by Ward's Method for SBDATA

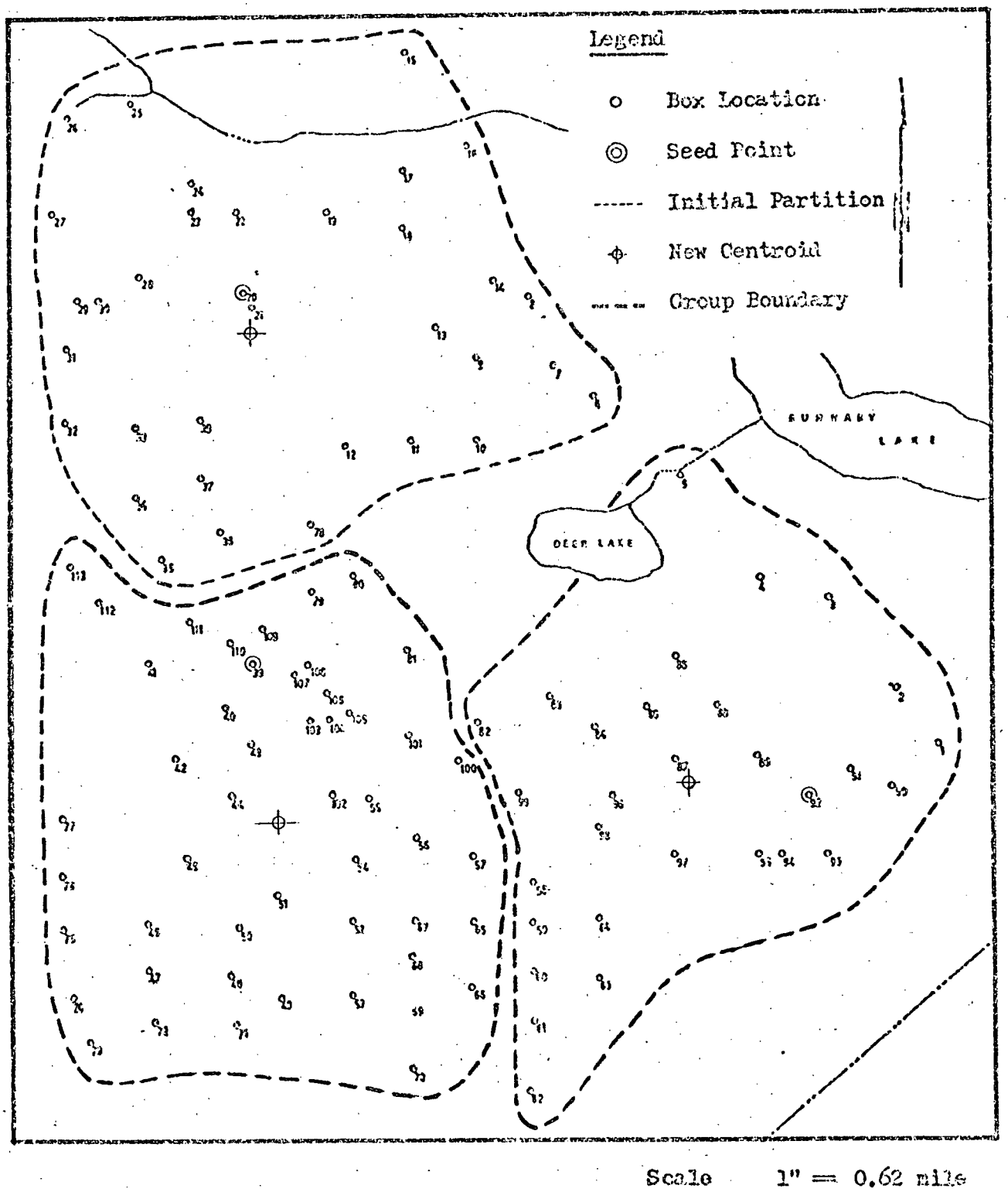


Figure 49. Group Boundaries Defined by Jancey's Method Using Seed Points as Inputs for SBDATA

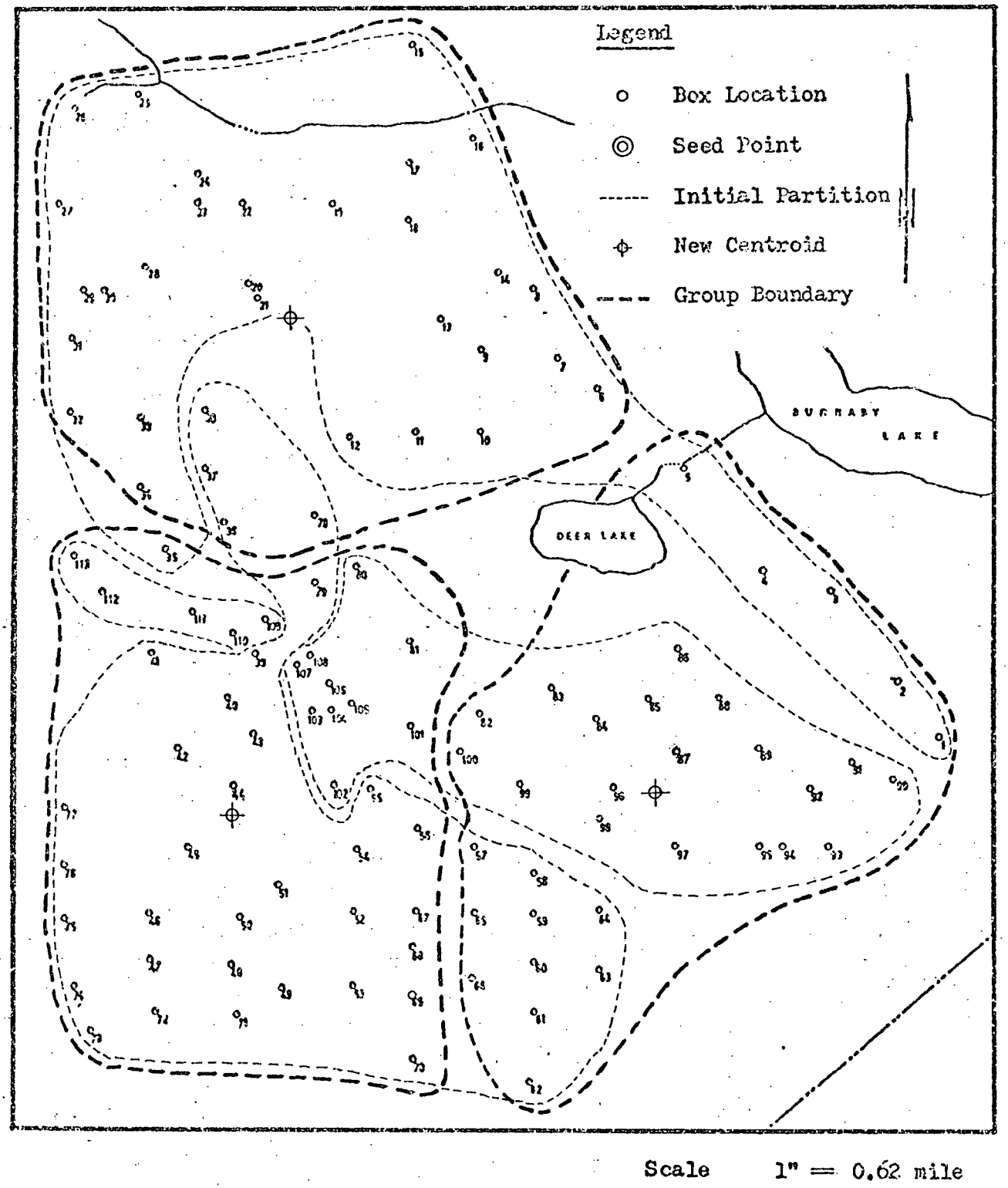


Figure 50. Group Boundaries Defined by Forgy's and Convergent K-mean Methods Using Initial Partitions for SBDATA

The first part of the program records the data points; calculates the distance among data points within the groups; computes the mean and standard deviations and plots a histogram of the distribution of distances (Appendix H). The distributions of distances within group are plotted because they are related to the route distance within each group. Christofides (1969) indicates that the associated distance for optimized truck route (D_0) is a function of radial distances (D_1) and the average value of the maximum number of customers that can be serviced by one route (C). These statistics are then compared and used as evaluation key for the methods.

The second part of the program was modified from ROUTPLOT, a computer program developed for plotting routes using maximum distance saving as criterion (Chance etal, 1975). This routine gives the travelling distances, the times required and the order of call points for the maximum saving route within the defined groups (Appendix H). Since timing and travelling distances are critical in the scheduling problems for the Post Office, these outputs are useful in judging the usefulness of clustering methods.

5.5 Evaluation

The discussion of the results in the previous sections have indicated that subjective evaluation of the results need to be supported by some objective measures. The last section introduced two appropiate tools for evaluating the group sizes quantitatively. Since the four sets of data each has spatial characteristics of its own, it is best to evaluate the methods for each data set separately. In the following subsections, the methods will be evaluated using the outputs from the two analysing tools.

5.5.1 Evenly distributed Contrived Data (DATA1)

As discussed in Section 5.3.1, the complete linkage, the Ward's and the three nonhierarchical methods each groups this data set into two groups of comparable sizes (Table 4). An examination of the plots of the groups defined by these methods (Figure 9, 13-15), however, does not indicate any distinct superiority of one method over the other. Both the radially linkage of hierarchical methods and laterally grouping of nonhierarchical methods have their own grouping boundaries. These groupings of comparable sizes are indifferent from theoretical standpoint, but in actual case, if there are restrictions imposed on the travelling directions, then the resulting group boundaries would be critical to the scheduling problem.

The outputs from the program ROUTE for this set of data as grouped by various methods are summarized in Table 11 and 12. These outputs are affected by the group sizes and the distribution of the data units within each group. Contrary to the group size comparisons, the complete linkage groupings do not have comparable means of distances or travel times and distances. This is the result of grouping scattered data units into clusters. The Ward's method groupings, though gives fairly similar means, do not have similar travelling distance or timings because of the unevenly scattering of data units within the defined groups. The three nonhierarchical methods give identical results because of the similarity of their groupings. The travelling distances and times for groupings of these methods, in this case, vary slightly and are the best matches among the 12 sets of results.

The distribution patterns of inter-data unit distances within groups are critical in the determination of travel distances and times. The patterns of the groups of the above five clustering methods, as shown in Figures 51-53, definitely reflect the similarity between the two groups defined by these methods. The patterns of the two groups defined by the complete linkage method are distinctly dissimilar. The Ward's patterns have better resemblance whereas that of nonhierarchical methods have almost identical distribution patterns. The degree of resemblance does not only offset the mean and standard

Methods \ Group	Mean (feet)		Std. Deviation	
	1	2	1	2
Hierarchical				
Single Linkage				
City - Block	20332	6791	10629	3413
Euclidean Distance	20332	6791	10629	3413
Chi-Squares	13281	19494	6499	10577
Complete Linkage	13926	16008	6757	7863
Avg. Linkage between Merged Group	16747	12115	8178	5977
Avg. Linkage within New Group	16747	12115	8178	5977
Centroid Method	16233	12800	7973	6008
Median Method	16233	12800	7973	6008
Ward's Method	15574	15004	7962	7671
Nonhierarchical				
Jancey's Method	18396	17811	10240	10353
Fogy's Method	18396	17811	10240	10353
Convergent K-mean Method	18396	17811	10240	10353

Table 11. Means and Standard Deviations of Groups Defined by 12 Cluster Methods for DATA1

Methods \ Group	Travel Time(min)		Travel Dist.(mi)		Total Time (min)	
	1	2	1	2	1	2
Hierarchical						
Single Linkage City - Block	202.82	27.92	50.70	6.98	263.12	39.62
Euclidean Distance	202.82	27.92	50.70	6.98	263.12	39.62
Chi-Squares	77.07	168.93	19.27	42.23	96.87	221.13
Complete Linkage	107.85	136.96	26.96	34.24	139.35	177.46
Avg. Linkage between Merged Group	153.19	78.02	40.44	19.51	199.09	102.32
Avg. Linkage within New Group	153.19	78.02	40.44	19.51	199.09	102.32
Centroid Method	145.31	127.16	36.33	24.82	189.41	127.16
Median Method	145.31	127.16	36.33	24.82	189.41	127.16
Ward's Method	137.20	113.58	34.20	28.39	175.90	146.88
Nonhierarchical						
Jancey's Method	120.50	118.83	30.12	29.71	152.90	158.43
Forgy's Method	120.50	118.83	30.12	29.71	152.90	158.43
Convergent K-mean	120.50	118.83	30.12	29.71	152.90	158.43

1. & 2. from first to last box ; 3. including stopping time.

Table 12. Travel Times and Distances of Groups Defined by 12 Cluster Methods for DATA1

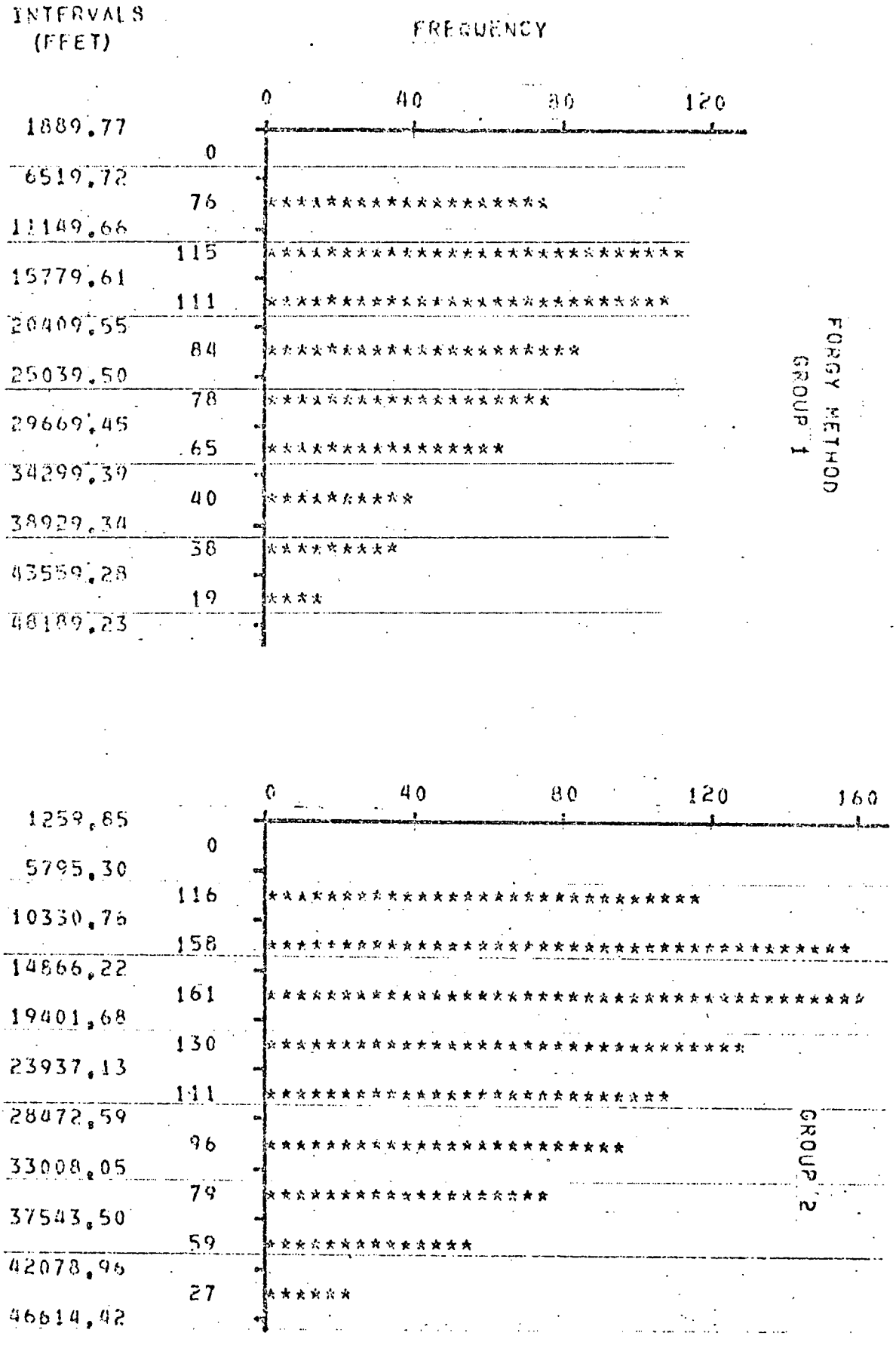


Figure 51. Distribution of Distances Within Groups Defined by Nonhierarchical Methods

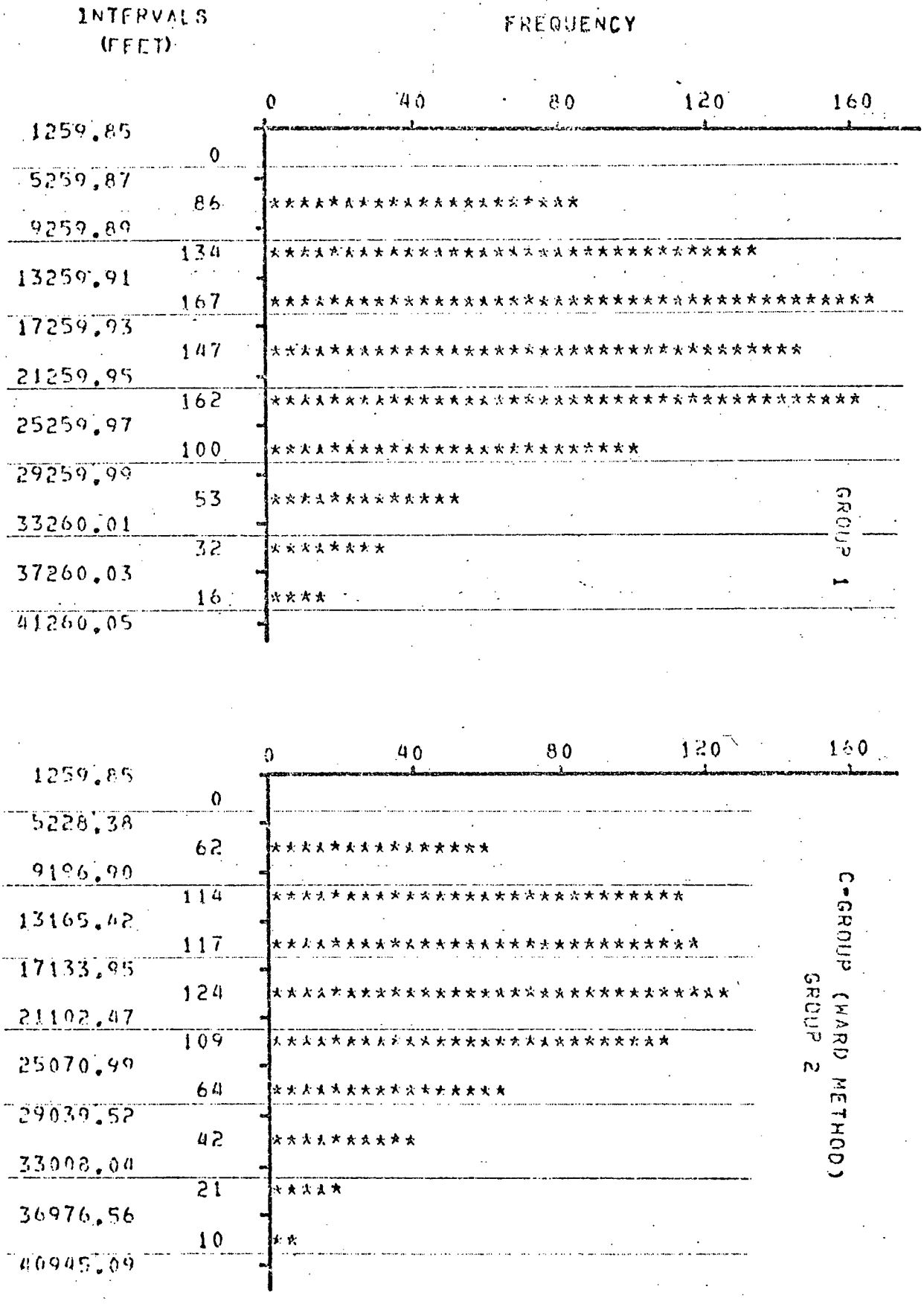


Figure 52. Distribution of Distances Within Groups Defined by Ward's Method for DATA1

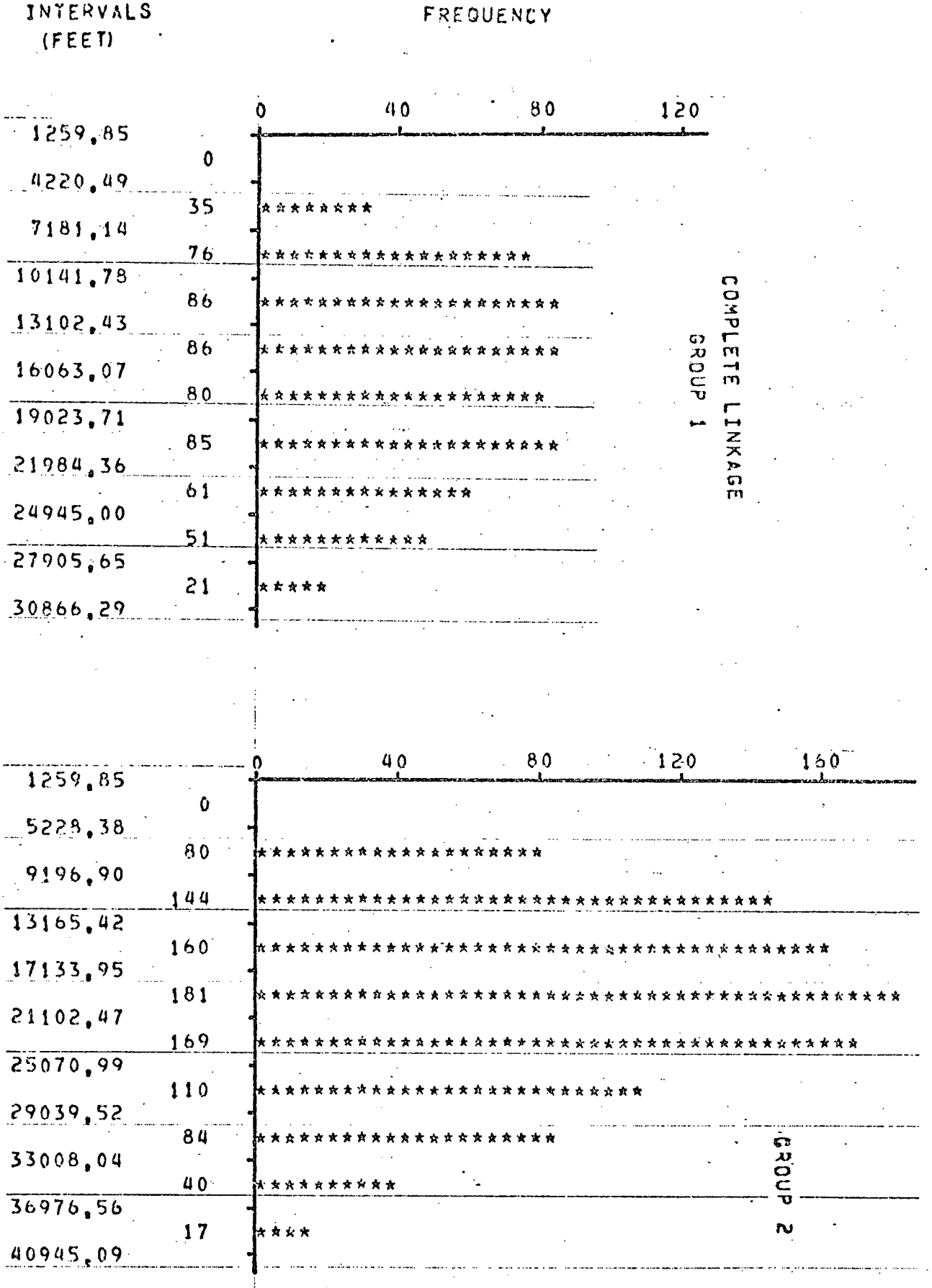


Figure 53. Distribution of Distances Within Groups Defined by Completer Linkage for DATA1

deviations of the distances, but also influence the travel distances and times.

Based on the above evaluations, the nonhierarchical methods' groupings definitely show better correlations in means, travel distances and times, and distribution patterns though there is little similarities in the group sizes. If there were travel restriction in lateral movements, then the Ward's method would probably be the best choice among these methods. On the whole, this quantitative evaluation of the grouping results indicates that the five methods mentioned in this sub-section are favoured for this set of data based on the assumption that there is no restrictions in travel directions.

5.5.2 Unevenly Distributed Contrived Data (DATA2)

The plots of the results of the 12 clustering methods indicate that the complete linkage and the two average linkage methods give results resembling to that of the intended group sizes (19, 34 and 27). A study of the summarized results (Table 13 and 14) from the program ROUTE for these sets of different group sizes and memberships also indicates that these three methods have very similar means of distances, and travel times. This is probably the result of various density of data units within each group. Group 1, the smaller and more scattered cluster has less stops, but longer

Methods \ Group	Mean (feet)			Standard Deviation		
	1	2	3	1	2	3
Hierarchical						
Single Linkage						
City - Block		N. A.			N. A.	
Euclidean Distance	5252	6471	14692	2414	3375	9592
Chi-Squares	5252	6471	14692	2414	3375	9592
Complete Linkage	6085	8237	8434	3131	4281	4574
Avg. Linkage between Merged Group	6642	8124	8434	3672	4285	4575
Avg. Linkage within New Group	6642	8124	8434	3672	4285	4575
Centroid Method	6642	9358	6754	3672	5040	3273
Median Method	6642	9358	6754	3672	5040	3273
Ward's Method	11668	7569	7083	7257	4573	3468
Nonhierarchical						
Jancey's Method	10424	14040	13645	8433	9152	8625
Forgy's Method	6955	13900	14644	4730	8609	9671
Convergent K-mean	6955	13900	14644	4730	8609	9671

Table 13. Means and Standard Deviations of Groups Defined by 12 Cluster Methods for DATA2

Methods \ Group	Travel Time[1](min.)			Travel Dist.[1](mi)			Total Time[2](min.)		
	1	2	3	1	2	3	1	2	3
Hierarchical Single Linkage									
City - Block		N.A.			N.A.			N.A.	
Euclidean Distance	25.29	34.85	74.45	6.32	8.71	18.61	40.59	57.34	108.65
Chi-Squares	25.29	34.85	74.45	6.32	8.71	18.61	40.59	57.34	108.65
Complete Linkage	30.06	50.58	50.82	7.52	12.65	12.71	47.16	81.18	75.12
Avg. Linkage between Merged Group	32.45	48.20	50.82	8.11	12.05	12.71	50.45	77.90	75.12
Avg. Linkage within New Group	32.45	48.20	50.82	8.11	12.05	12.71	50.45	77.90	75.12
Centroid Method	32.45	65.38	40.09	8.11	16.34	10.02	50.45	98.68	60.79
Median Method	32.45	65.38	40.09	8.11	16.34	10.02	50.45	98.68	60.79
Ward's Method	69.67	26.72	42.47	17.42	6.68	10.62	107.47	39.32	64.07
Nonhierarchical Jancey's Method	45.27	60.61	69.67	11.39	15.15	17.42	66.27	81.31	100.27
Forgy's Method	31.97	66.81	67.05	7.99	16.70	16.76	46.37	92.91	98.55
Convergent K-mean	31.97	66.81	67.05	7.99	16.70	16.76	46.37	92.91	98.55

1. from 1st to last box; 2. including stop time.

Table 14. Travel Distances and Times of Groups Defined by 12 Cluster Methods for DATA2

distances between stops in travelling from one location to another. Group 2 and 3 have denser population within the groups, thus requiring more stops in traversing the shorter inter-point distances. These characteristics are reflected in the distance distribution patterns (Figures 54-55) of the above three linkage methods.

The intended group sizes for the three clusters are not of comparable number of members and this makes the evaluation of these results difficult. Undoubtly, the evaluation of these outcomes must be subjective. If similar travel distances among the three groups are required, then the Jancey's nonhierarchical method has groups of this nature. On the other hand, if both the travel times and the mean of distances are to be comparable, then the complete linkage and the two average linkage methods would be more appropiate in grouping this set of data. One definite conclusion, however, can be drawn in the evaluation of these clustering techniques is that single linkage using " City - Block " measure is not an appropiate method for this set of data. Among the other seven methods, the two centroid methods also have similar group sizes as the intended ones. However, the results from ROUTE for the groupings of these methods do not indicate any distinct characteristics that are more superior than the three linkage methods' and the Jancey's method's. The other five methods do not group data units into clusters with sizes resembling to the

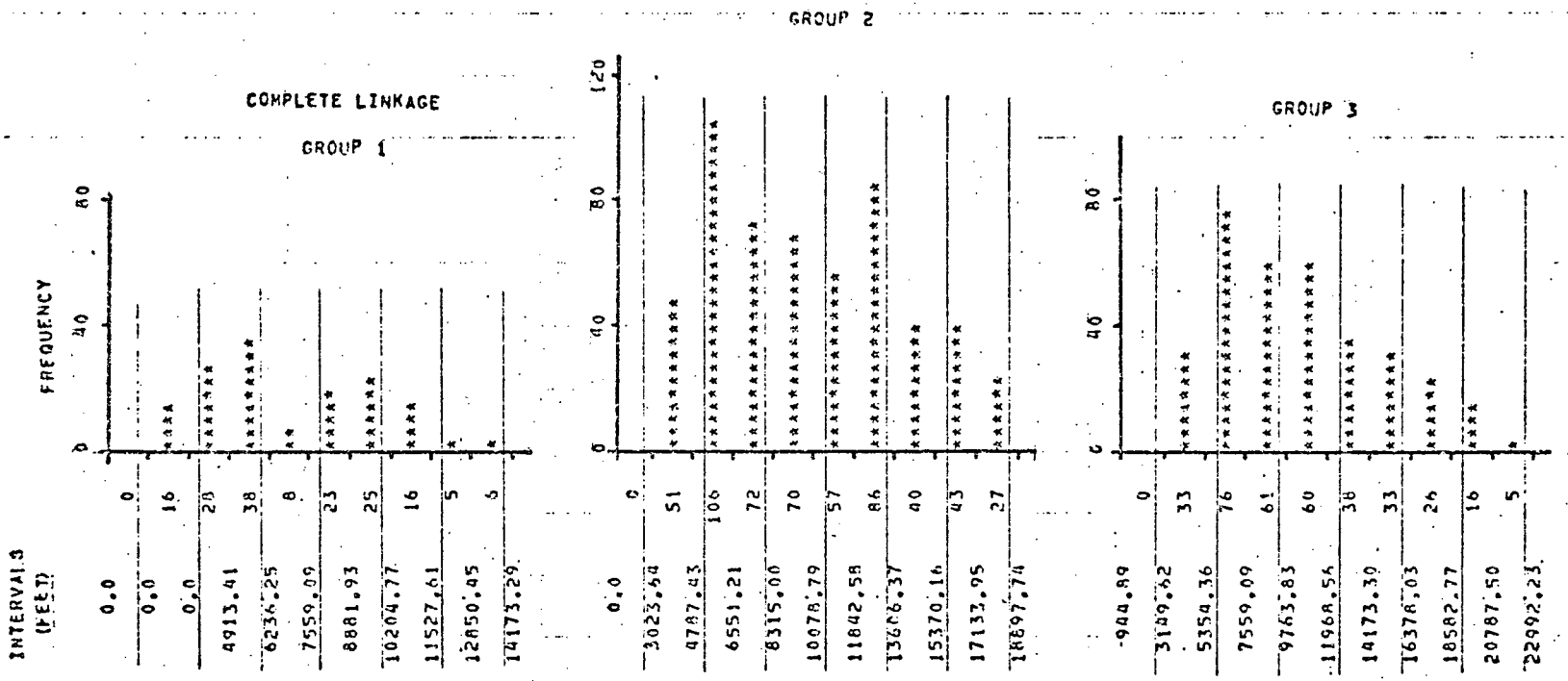


Figure 54. Distribution of Distances Within Groups Defined by Complete Linkage for DATA2

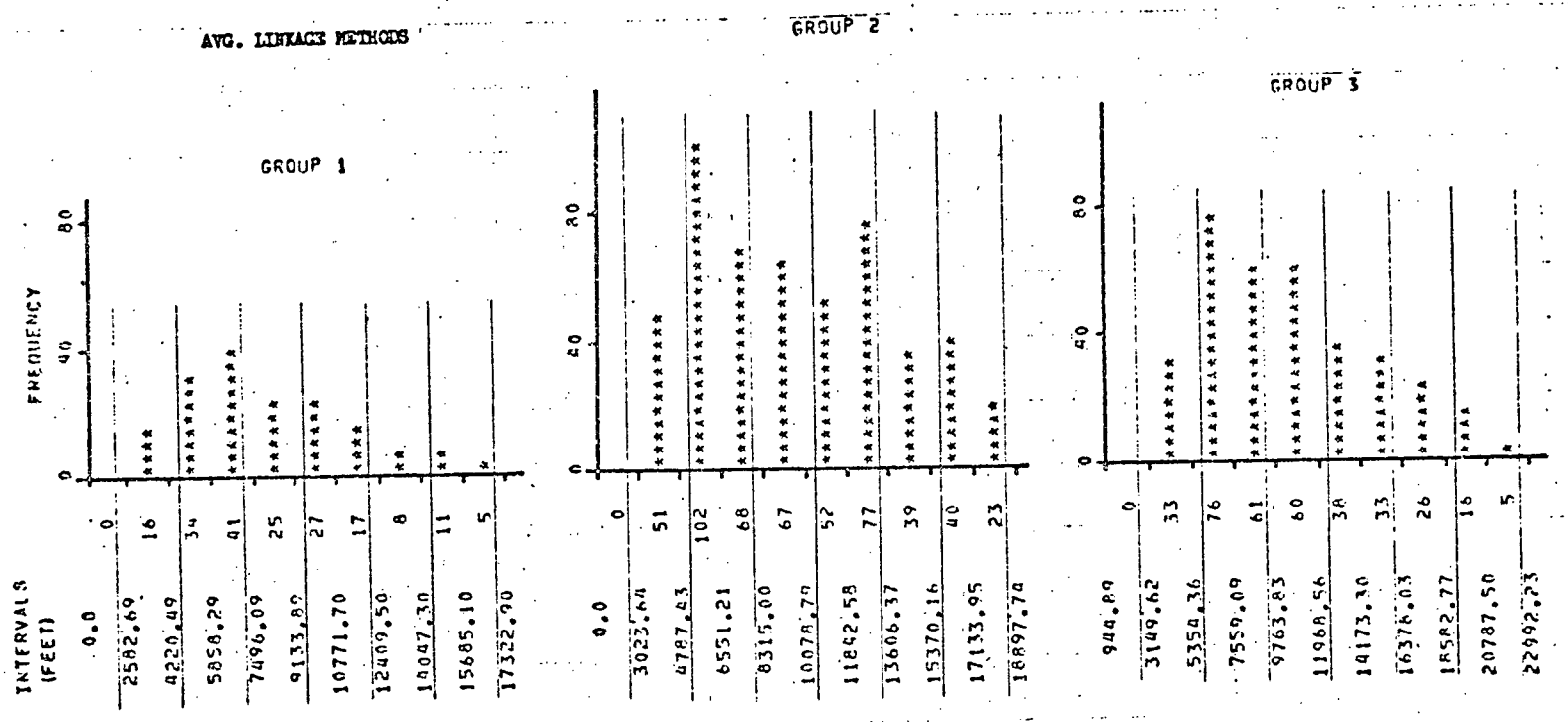


Figure 55. Distribution of Distances Within Groups Defined by Avg. Linkage Methods for DATA2

intended groupings nor have comparable means of distances and travel distances within the defined groups.

On the whole, there are methods that are more suitable for grouping this set of data. The superiority of any method, in this case, cannot be judged solely by the comparisons of the means of distances, travel distances and times; it requires subjective evaluation. This subjective decision process involves the consideration of appropiate group sizes, the weighing of the quantitative measures, and most important of all, the experience of the user.

5.5.3 North Burnaby Empirical Data (NBDATA)

The complete linkage, the average linkage within the new group, the Ward's and the Jancey's methods all have similar group sizes though different memberships. The uneven distribution of data points, especially points 1, 2 and 3, in this set do not only discourage the use of group sizes as an evaluation of criterion, it also indicates the importance of distribution pattern in the judgement of appropiate groupings.

The application of evaluation tools on the 12 sets of groupings defined by the different clustering methods on this data set indicates that there are conflicting evidence on the superiority of one method over the others (Tables 15 and 16).

Methods \ Group	Mean (feet)		Std. Deviation	
	1	2	1	2
Hierarchical				
Single Linkage				
City - Block	944	11637	476	7241
Euclidean Distance	944	11637	476	7241
Chi-Squares	9562	7705	5875	4112
Complete Linkage	9824	6501	5609	3215
Avg. Linkage between Merged Group	944	11637	476	7241
Avg. Linkage within New Group	8614	10087	8105	6125
Centroid Method	9489	9531	7226	5293
Median Method	9489	9531	7226	5293
Ward's Method	10269	7622	6187	4201
Nonhierarchical				
Jancey's Method	9749	6688	5553	3332
Forgy's Method	9576	7091	5487	3588
Convergent K-mean Method	9576	7091	5487	3588

Table 15. Means and Standard Deviation of Groups Defined by 12 Cluster Methods for NBDATA

Methods \ Group	Travel Time[1](min)		Travel Dist.[1](mi)		Total Time[2](min)	
	1	2	1	2	1	2
Hierarchical Single Linkage						
City - Block	1.00	159.91	0.25	39.98	3.70	235.51
Euclidean Distance	1.00	159.91	0.25	39.98	3.70	235.51
Chi-Squares	50.06	63.25	12.51	15.81	76.16	115.45
Complete Linkage	66.81	49.73	16.70	12.43	100.11	94.73
Avg. Linkage between Merged Group	1.00	159.91	0.25	39.98	3.70	235.51
Avg. Linkage within New Group	70.68	66.19	17.67	16.55	105.78	109.39
Centroid Method	21.21	87.83	5.30	21.96	32.01	155.33
Median Method	21.21	87.83	5.30	21.96	32.01	155.33
Ward's Method	68.19	57.55	17.05	14.39	99.69	104.35
Nonhierarchical Jancey's Method	65.00	54.07	16.25	13.52	96.50	100.87
Forgy's Method	59.46	60.15	14.87	15.04	87.36	110.55
Convergent K-mean	59.46	60.15	14.87	15.04	87.36	110.55

1. from 1st to last box; 2. including Stopping time

Table 16. Travel Distances and Times of Group Defined by 12 Cluster Methods for NBDATA

The centroid methods' groupings have the most similar means of distances, whereas the Forgy's and the Convergent K-mean nonhierarchical methods' clusters have the almost identical travel times and distances. This contrast in the quantitative measures is the result of the differences in the group sizes as well as the distribution pattern of the data points within the groups.

In this case, the most similar group sizes (39, 48) as defined by the average linkage within the new group method does not have the most matched travel times or distances. The inclusion of a potential cluster of 3 points (1, 2 and 3) in the groupings distorts the distribution pattern of the distances (Figures 56-58). The resultant travel time required for traversing a smaller, but fairly scattered group is relatively similar to that required for travelling through the points of a denser population. The two nonhierarchical methods (Forgy's and Convergent K-mean) have groupings of comparable distance distributions, thus the resulting travel times and distances are similar. These two clustering methods are probably the most suitable methods among the 12 for this data set.

Similar to the conclusion drawn for DATA1, the single linkage methods do not group data points into groups of comparable sizes, nor similar travel distances. The

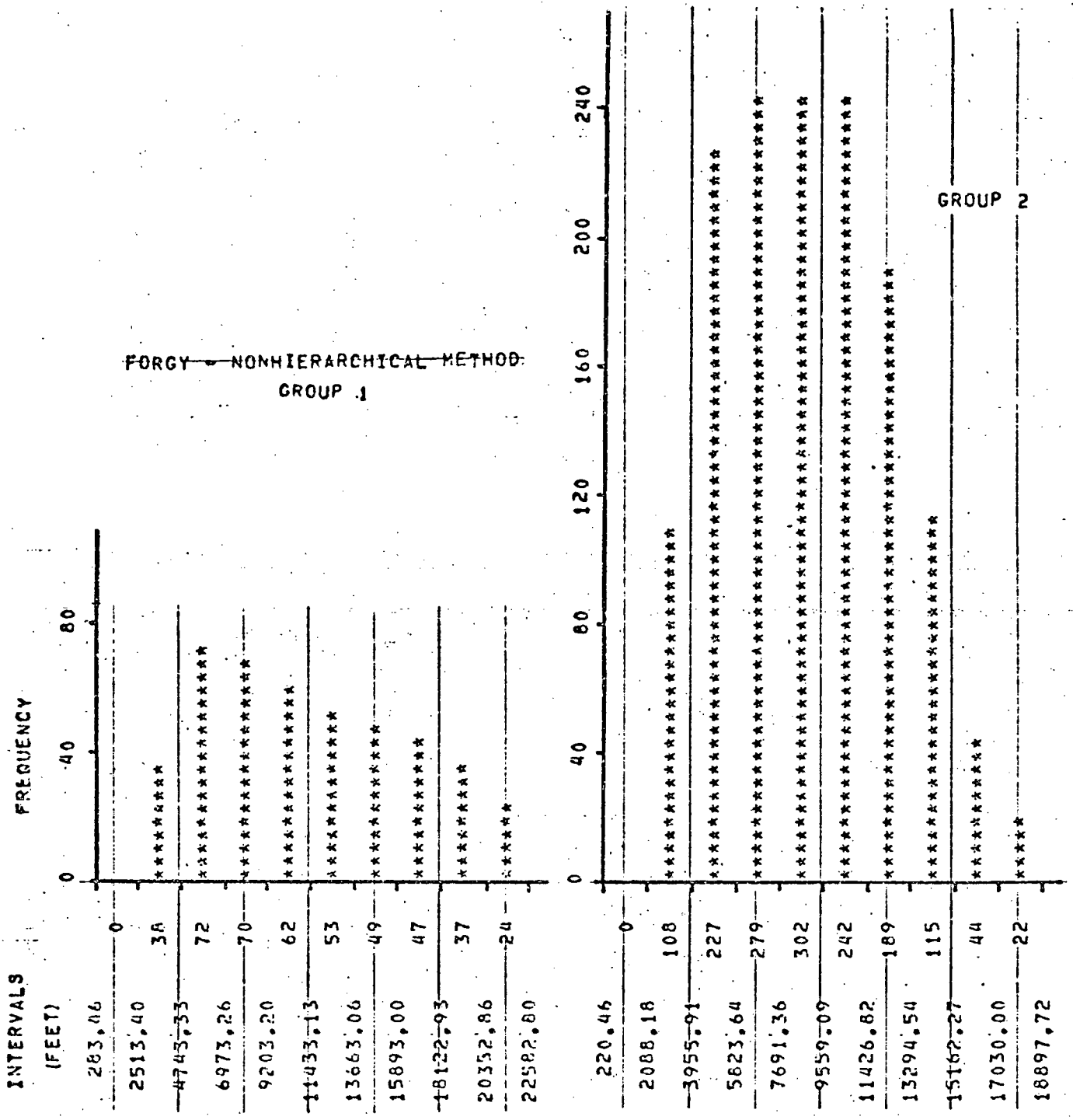


Figure 56. Distribution of Distances Within Groups Defined by Forgy's and Convergent K-mean Methods for NBDATA

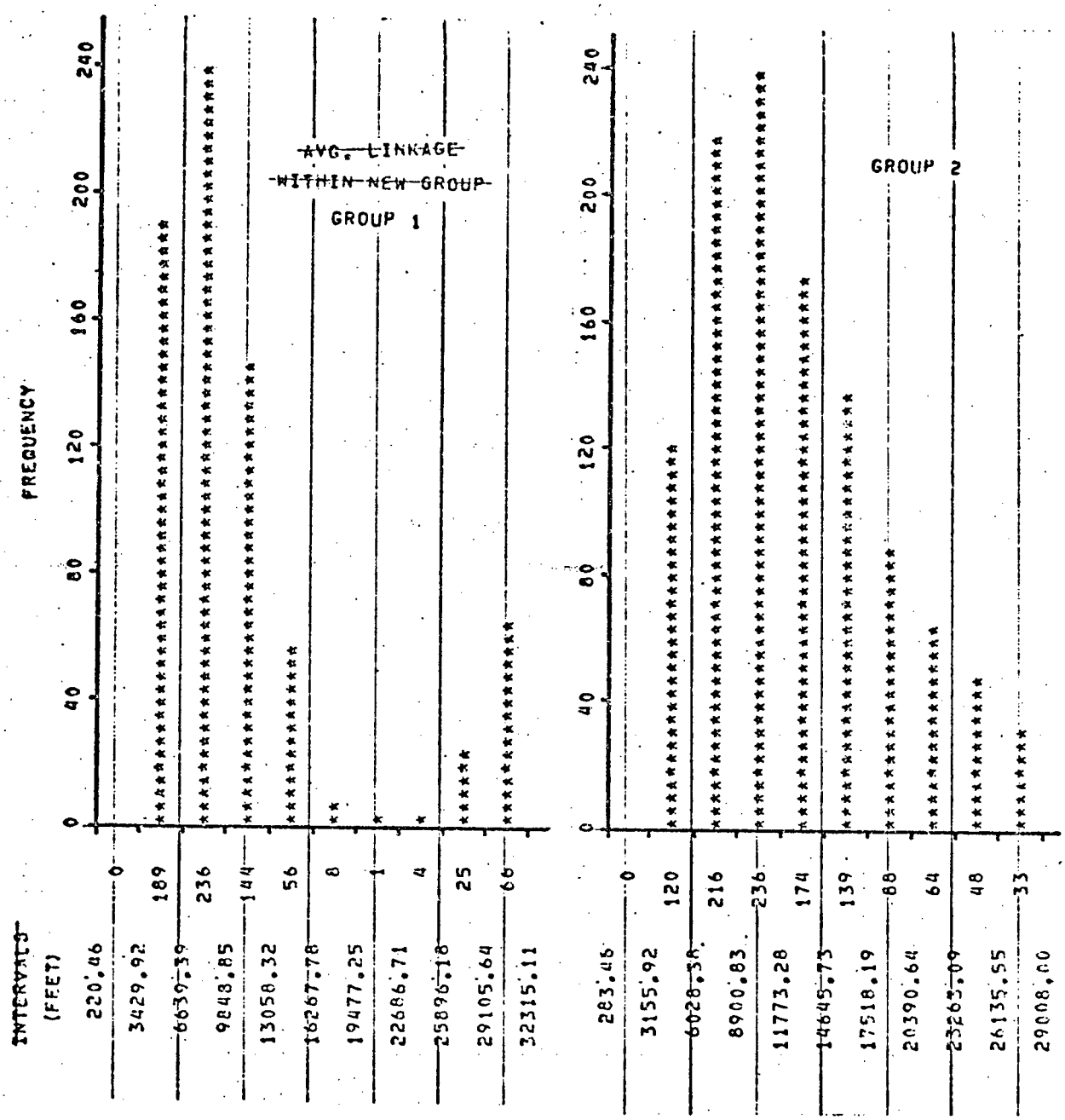


Figure 57. Distribution of Distances Within Groups Defined by Average Linkage Methods for NBDATA

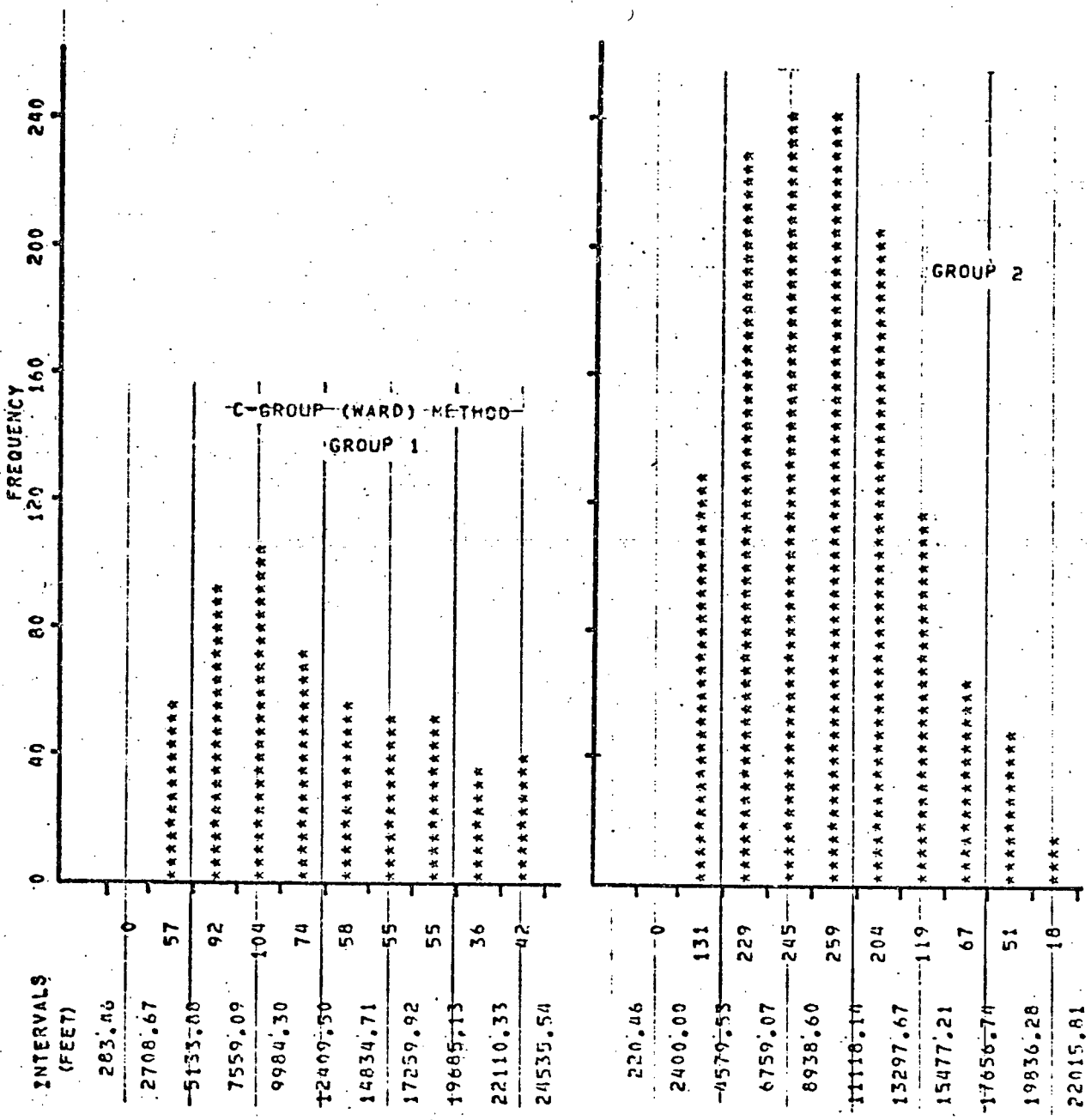


Figure 58. Distribution of Distances Within Group Defined by Ward's Method for NBDATA

average linkage between merged groups method's clusters also do not have similar sizes nor travel times. These methods are inappropiate because they tend to isolate the potential 3-point cluster from the rest of the data units. This isolation yields drastically different group sizes and, of course, totally incompatible travel times. The two centroid methods have similar clustering properties though less severe in the degree of isolation of the three data units. The results from the other four methods are generally acceptable but not recommended for clustering this set of data.

5.5.4 South Burnaby Empirical Data (SBDATA)

The oddly clustered group sizes for single linkage (1, 1 and 111) and the centroid methods (2, 18, 93) were not examined in this evaluation process. These methods have failed to cluster the data set into groups of similar sizes and are ruled out as aids for clustering this set of data for the scheduling problem. The summary table of the group sizes (Table 9) indicates that the nonhierarchical methods and the single-linkage chi-squares techniques cluster the data set into three comparable groups.

An examination of the results from ROUTE on the twelve sets of groupings as defined by the various methods indicates that only three methods' groupings actually have

comparable means of the distances, travel times and distances (Tables 17 and 18). The Jancey's method groups have distinctly similar travel times as well as total times for the traverse within the groups: a result in grouping the data points into regions of almost identical areas (Figure 49). The similarity of distribution patterns of the inter-unit distances for groups as defined by this method (Figure 59) also reflects the superiority of this method over the others.

Among the hierarchical methods, only the single-linkage Chi-squares method and the Ward's method have comparable means of distances as well as travel times within the defined groups. The group sizes for the Ward's method are 45, 28 and 40 respectively (Figure 47). Though the scattered distributions of all the data units within these groups resemble each other, the distances between points for group 2 (28 points) are slightly longer than that of the other two groups. Thus the distribution of distances (Figure 60) has a wider spread than the others. The number of stops is, in this case, critical for the total travel and stop time.

The single linkage Chi-square method has its distince linkage pattern (Figure 41). Although the group sizes are 34, 36 and 43 respectively, the defined boundaries are very different from that of any methods examined in this

Methods \ Group	Mean (feet)			Standard Deviation		
	1	2	3	1	2	3
Hierarchical						
Single Linkage						
City - Block		N. A.			N.A.	
Euclidean Distance	5009	6937	7766	2280	3541	4016
Chi-Squares	7919	7712	7784	5011	4896	3697
Complete Linkage	4738	6566	8309	2107	3231	4029
Avg. Linkage between Merged Group	5009	6037	7766	2280	3541	4016
Avg. Linkage within New Group	5146	6230	8283	2350	3232	4132
Centroid Method	N.A.	5603	9949	N.A.	2993	4975
Median Method	N.A.	5603	9949	N.A.	2993	4975
Ward's Method	7039	7346	6333	3818	3847	3348
Nonhierarchical						
Jancey's Method	7099	6345	6465	3328	3180	3336
Forgy's Method	7169	6470	6269	3363	3205	3210
Convergent K-mean	7169	6470	6269	3363	3205	3210

Table 17. Means and Standard Deviations of Groups Defined by 12 Cluster Method for SBDATA

Methods \ Group	Travel Time[1] (min)			Travel Dist.[1] (mile)			Total Time[2] (min.)		
	1	2	3	1	2	3	1	2	3
Hierarchical Single Linkage									
City - Block		N.A.			N.A.			N.A.	
Euclidean Distance	27.75	39.23	77.76	6.94	9.81	19.44	46.65	62.63	137.16
Chi-Squares	38.65	50.43	61.97	9.66	12.67	15.49	69.25	82.83	100.67
Complete Linkage	24.19	50.14	71.48	6.05	12.54	17.87	42.19	90.64	114.68
Avg. Linkage between Merged Group	27.75	39.23	77.76	6.94	9.81	19.44	46.65	62.63	137.16
Avg. Linkage within New Group	28.91	40.84	75.88	7.23	10.21	18.97	48.71	73.24	125.38
Centroid Method	N.A.	27.97	190.16	N.A.	6.99	47.75	N.A.	44.17	273.86
Median Method	N.A.	27.97	190.16	N.A.	6.99	47.75	N.A.	44.17	273.86
Ward's Method	57.97	44.24	48.74	14.49	11.06	12.18	98.47	69.44	84.74
Nonhierarchical Jancey's Method	49.22	54.19	48.96	12.30	13.55	12.17	78.92	95.39	79.29
Forgy's Method	50.78	56.12	35.90	12.70	14.02	8.97	98.47	69.44	62.90
Convergent K-mean	50.78	56.12	35.90	12.70	14.02	8.97	98.47	69.44	62.90

1. from 1st to last box; 2. including stopping time.

Table 18. Travel Distances and Times of Groups Defined by 12 Cluster Methods for SBDATA

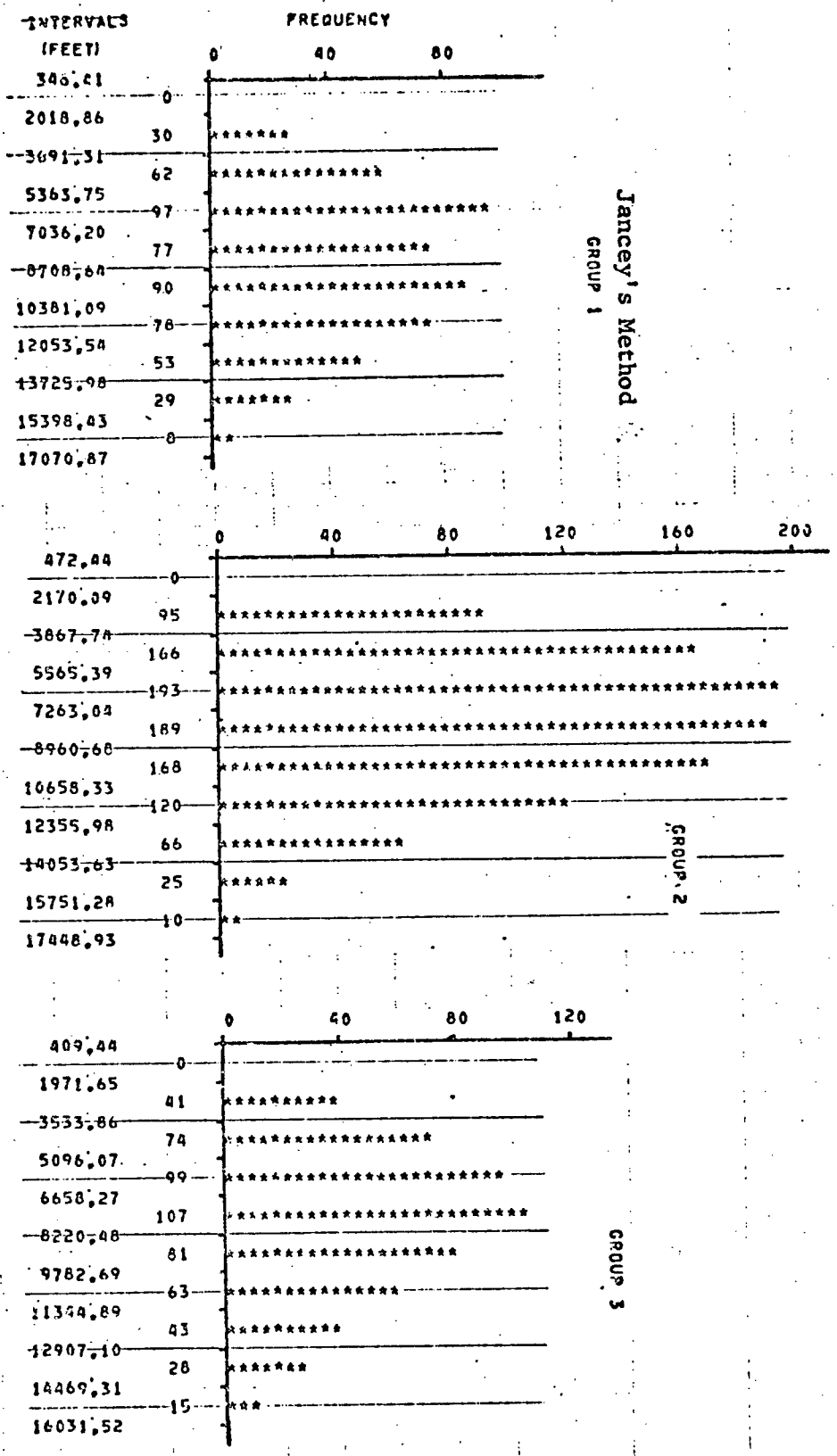


Figure 59. Distribution of Distances Within Groups Defined by Jancey's Method for SBDATA

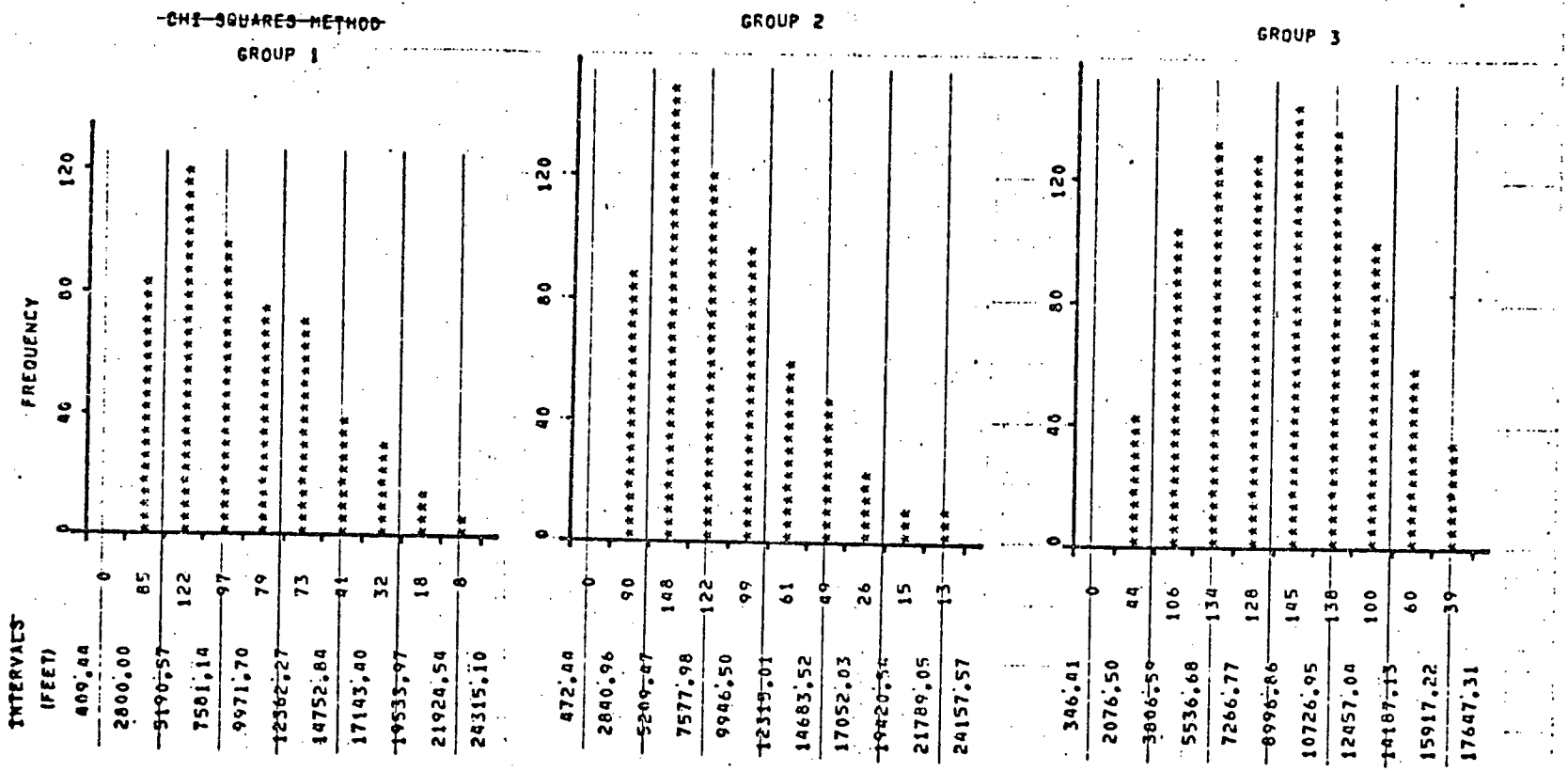


Figure 60. Distribution of Distances Within Groups Defined by Chi-Squares Method for SBDATA

study. The distribution pattern of the distances is fairly similar to the Ward's. These distributions are slightly skewed to the left as shown in Figure 61. The skewness is parallel to the difference in the means and standard deviation of the distances. The resulting route travel time and distance also differ slightly (Table 18), because of the different degree of scattering of the data units within the different sized groups.

The differences in population density of the groups, as defined by the above three methods, plays an integral part in the calculation of distances and the choice of route. The nonhierarchical (Jancey's) method is apparently the best clustering techniques among the twelve methods for grouping this set of data.

5.6 Summary

As mentioned in the above sections, evaluation of the applicability of the twelve clustering methods is difficult and has to be very subjective. In section 5.5, the quantitative evaluation approaches are designed to aid the evaluation process. It is found that not all the methods are suitable for all kinds of data sets with various spatial chatacteristics. This conclusion is parallel to that drawn in section 5.3.

Figure 61. Distribution of Distances within Groups Defined by Ward's Method for SBDATA

The choice of clustering method for grouping data units depends highly on the characteristics of the data set. A table (Table 19) summarizing the preference of methods based on the group sizes, the data unit distributions within the groups, and the travel times and distances required for traversing the data points within the groups indicates that various methods are favoured for clustering different data sets. Methods that give appropiate groupings for the evenly distributed data sets (DATA1, NBDATA and SBDATA) do not necessarily give satisfactory clusters for other data sets. Among these methods that are considered applicable to the scheduling problem, some are higher ranked than the others for one set of data, and vice-versa.

The spatial characteristics of the data units within a group, perhaps, is the most critical element in the decision process. DATA1, a data set without any potential cluster, is best grouped by the three nonhierarchical methods. The results of these methods all have similar characteristic: the data units are all clustered into lateral groups. This characteristic could be one of the deciding factor in the evaluation process. The inclusion of a potential 3-point cluster in the north east corner of the area in NBDATA reduces the effectiveness of some methods. Only two hierarchical methods are favoured for their resulting travel distances

Ranking Criterion	Group Sizes				Travel Time & Dist.			
Methods \ Data Set	1	2	3	4	1	2	3	4
Hierarchical								
Single Linkage								
City - Block								
Euclidean Distance								
Chi-Squares				1				3
Complete Linkage	3	1	2		3	1		
Avg. Linkage between Merged Group		2				2		
Avg. Linkage within New Group		2	1			2	2	
Centroid Method								
Median Method								
Ward's Method	2		3		2		3	2
Nonhierarchical								
Jancey's Method	1			3	1			1
Forgy's Method	1			2	1		1	
Convergent K-mean	1		3	2	1		1	

Legend

1 — best method

2 — second best method

3 — third best method

Table 19. Summary of Method Preferences for the Four Sets of Data

and times. The Ward's method is considered to be the alternative technique for grouping this data set. SBDATA, a data set similar to DATA1, is best grouped by Jancey's nonhierarchical method. On the whole, the three nonhierarchical methods are generally ranked as applicable methods for grouping evenly distributed data sets. The most ranked hierarchical method is the Ward's technique though the results from this method are less appealling than the nonhierarchical ones'.

The unevenly distributed data set with intended grouping is found to be best grouped by the three linkage methods: the complete linkage and the two average linkage methods. Unfortunately, there were no available empirical data that resemble this DATA2 set, and the conclusions concerning the clustering of this type of data distribution can only be drawn from the trials on DATA2. It is difficult to state that these three methods are the best ones for grouping this type of distribution, and it requires supporting evidence to substantiate the validity of this subjective evaluation.

In evaluating the clustering methods for grouping mail boxes for the scheduling problem, the ranking system based on the similarity of travel distances and times is probably the best approach. In most cases, the schedules for the

truck services are set up according to the time allocated to each driver and not to the number of boxes each driver has to service. The Postal Union ruling on the maximum number of stops for each route, however, could hinder the use of above scheduling approach. For this study, it is assumed that the schedules are prepared for each driver based on the time slot assigned to each type of services. If the assumption is correct, then the three nonhierarchical methods and the Ward's method are probably the better clustering techniques for grouping data sets for Burnaby area.

CHAPTER VI

CONCLUSIONS

This study in investigating the applicability and characteristics of the twelve clustering techniques indicates the following:-

1. The spatial characteristics of the data set or the locations of the mail boxes have significant influence on the outcomes of the grouping methods. A set of fairly evenly distributed data units could be best grouped by the nonhierarchical methods and not the others. Conversely, these methods are not suitable for grouping unevenly distributed data units. There is no straight rule, however, to discriminate the use of any clustering method, and it requires test runs to prove the suitability of these clustering methods for the particular data set.

2. The three single linkage methods using different distance measures give various results for the same set of data. The method using simple "City - Block"as association measure tends to

group closely located data units into a cluster and isolate the more distant data points. This grouping characteristic generates dissimilar group sizes as well as travel distances and times for traverses within defined groups. The square root of the sum of squares of differences (Euclidean) method does not always isolate the distant data units from the others, and this approach generally gives fairly acceptable groupings of the natural clusters. The chi-squares method clusters data units with distinctive links, and the group sizes outlined by this method is usually acceptable, but not as good as other methods'. On the whole, single linkage methods are not suitable techniques for grouping these four sets of data.

3. The two average linkage methods and the complete linkage technique are considered to be suitable only for the unevenly distributed data set. The criteria used in these methods differ from that of single linkage methods', and these grouping criteria generally link data units into groups of fairly comparable sizes. These algorithms tend to link the potential clusters together prior to the linking of more scattered points. The effectiveness of the ability to group potential clusters is highly reduced if there is an

absence of potential groupings. The results from these methods for evenly distributed data sets are therefore not as satisfactory as these of nonhierarchical and the Ward's methods'. On the other hand, intended groupings are readily clustered by these methods as shown in the trials for DATA2.

4. The results generated by the centroid and median methods are always identical for these four data sets. The differences in weighing the inter-unit or inter-cluster diatances in these two methods have little influence on the linking sequence or the linkage pattern. It is apparent that only one of these methods should be used in further studies of the applicability of cluster analysis. The results produced by these methods do not conform to any of the better ranked methods, and they are not suitable for grouping the mail boxes for the scheduling problem.

5. The Ward's method is by far the best ranked among the seven hierarchical methods for grouping evenly distributed data sets. The error sum of squares criterion adopted in this method links the variables with large variances first, thus gives a good

representation of both the closely located and distant data units in the groupings. The results of the Ward's method for the unevenly distributed data set, however, is unacceptable.

6. The results from the nonhierarchical techniques are distinctly satisfactory for grouping evenly distributed data sets. The location of the seed points or initial partitions for these methods, contrary to the findings of other authors, is found to be of minor importance in the grouping of the evenly distributed data units. The continuous allocation and reallocation of the data units to the nearest centroid apparently reduces the importance of the seed points and partitions. In this study, randomly selected seed points and initial partitions are seemingly valid. This does not only reduce the complexity in using nonhierarchical methods, it also reduces the time used in data preparations.

7. The distribution of inter-unit distances within groups identifies the scattering of the data units. The similarity of distribution patterns would indicate the compatibility of travel distances and times required for the groups defined. This

conclusion is similar to the relationship between travel distances and the area in which the call points are located described by Christofides (1969). The comparison of the distribution pattern is definitely a useful aid in the selection of clustering methods.

8. The best interpretation tool for both the hierarchical and nonhierarchical clustering results is the representation of the links between entities or the group boundaries on a 2-dimensional graph. Tree diagrams are also useful, but it requires more time to trace a tree than to inspect the linkage or boundary plots. The plot of data units on graph also helps to understand the degree of scattering of the call points. On the whole, visual aids are useful in the preparation of schedules.

9. The travel distances and times are critical measures in evaluating the groupings defined by various methods. These elements are actually the most vital information in the preparation of schedules as well as the specific routes for the trucks. The optimal route as outlined by the maximum distance saving routing method for each

group can also be used as a decision criterion in the selection of clustering method.

10. The evaluation of clustering methods, whether qualitative or quantitative in nature, has to be very subjective and heuristic.

The seemingly apparent choice of methods for clustering mail boxes should be the Ward's and the three nonhierarchical methods for data points distributed similar to North and South Burnaby's. This, however, does not imply that all the groupings of the boxes into clusters should be performed by these approaches. It is expected that some areas, such as the North Shore, would warrant the use of other clustering methods, e.g. the complete linkage and the two average linkage methods, in order to give satisfactory groupings of the mail boxes. The tests on unevenly distributed data set DATA2 have shown that nonhierarchical methods are not suitable for this type of data point distribution.

The tools developed for this study in clustering data sets, calculating statistics, and estimating the route distance and timing could be coordinated into a single program for scheduling purposes. These tools are efficient means to help analyse the route structure as well as the spatial relationships of call points. The histograms showing

distribution of distances between points is also a useful tool for analysing the data points and evaluating the groupings.

This study also points out that although computerized clustering methods can help the schedulers in determining the assignment of call points, it does not over-rule the superiority of the groupings outlined manually by inspection as carried out by the experienced planners. An interesting study related to this clustering method investigation would be the study of applicability of the five more suitable methods in grouping the Vancouver's mail or bundle boxes into clusters. This trial would further prove the feasibility of using cluster analysis as an aid to the Post Office scheduling problem.

FOOTNOTE

1. Unpublished Special Vehicle Utilization Study Report, Vancouver Post Office, 1975.

2. M.R. Anderberg. Cluster Analysis for Application (New York: Academic Press, 1973). pp.25-29.

3. B. Everitt. Cluster Analysis (Toronto: Heinmann Educational Books, 1974). pp.7-9.

4. M.R. Anderberg. pp.132-133.

5. B.S. Duran and P.L. Odell. Cluster Analysis, a Survey (New York:Springer-Verlag, 1974). pp.6-7.

6. M.R. Anderberg. pp.136-7.

7. Ibid. p.134.

8. Ibid. p.140.

9. Ibid. p.156.

10. Ibid. p.163.

REFERENCES

Anderberg, M.R., Cluster Analysis for Application, Academic Press, N.Y., 1973, 354 p.

Astrahan, M.M., "Speech Analysis by Clustering, or the Hyperphoneme Method", Stanford Artificial Intelligence Project, Stanford University, Stanford, Calif., 1970, 25p.

Ball, G.H. and Hall, D.J., "A Clustering Technique for Summarizing Multivariate Data", Behavioral Sciences, vol. 12, No. 2, 1967, pp. 153-55.

Bijnen, E.J., Cluster Analysis: Survey and Evaluation of Techniques, Tilburg University Press, The Netherlands, 1973, 112p.

Bonner, R.E., "On Some Clustering Techniques", IBM Journal of Research and Development, 1964, vol.8, pp. 22-32.

Bridges, C.C., "Hierarchical Cluster Analysis", Psychological Reports, vol. 18, 1966, pp.851-54.

Carmichael, J.W., George, J.A. and Julius, R.S., "Finding Natural Clusters", Syst. Zool., vol.17, 1968, pp. 144-150.

Cattell, R.B., Factor Analysis, Harper, N.Y., 1952, p.355.

Chance, R., Dyke, B. and Wong, S., Unpublished Report on Special Vehicle Utilization Study, Vancouver Post Office, Vancouver, 1975, 40p.

Christofides, N. and Eilon, S., "Expected Distances in Distribution Problems", Op. Res. Q., vol. 20, no. 4, 1969, pp. 437-43.

Cochran, W.G. and Hopkins, C.E., "Some Classification Problems with Multivariate Qualitative Data", Biometrics, vol.17, no.1, 1961, pp.10-32.

Cole, A.J. and Wishart, D., "An Improved Algorithm for the Jardine-Sibson Method of Generating Overlapping Clusters", The Computer Journal, vol.13,1970, pp.156-163.

Cormack, R.M., "A Review of Classification", Jour. R. Statist. Soc. Series A, v. 134, no.3, 1971, pp. 321-367.

Duran, B.S. and Odell, P.L., Cluster Analysis, a Survey, Springer-Verlag, N.Y., 1974, 137p.

Edwards, A.W.F. and Cavall-Sforza, L.L., "A Method for Cluster Analysis", Biometrics, v.21, 1965, pp.362-375.

Engelman, L. and Fu,S., BMDP2M program, UCIA BMD Documentation, UCIA, Calif., 1970, 7p.

Everitt, B., Cluster Analysis, Heinmann Educational Books, London, 122p.

Fleiss, J.L. and Zubin,J., "On the Methods and Theory of Clustering", Multivariate Behavioral Research, v.4, 1969, pp.235-250.

Forgy, E., "Cluster Analysis of Multivariate Data: Efficiency vs Interpretability of Classifications", Biometrics, v.21, 1965, p.758.

Gower, J.C., "Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis", Biometrika, v.53, 1966, pp. 325-338.

_____, "A Comparison of Some Methods of Cluster Analysis", Biometrics, v.23, no.4, 1967, pp.623-37.

_____, "Minimum Spanning Trees and Single Linkage Cluster Analysis", Appl. Statist., v.18, no.1, pp.54-64.

Green, P.E. and Rao, V.R., "A Note on Proximity Measures and Cluster Analysis", Jour. Mark. Res., vol. VI, 1969, pp.359-64.

Harris, B., Farhi, A. and Dufour, J., Aspects of a Problem in Clustering, University of Pennsylvania, 1972, 28p.

Hartigan, J.A., "Representation of Similarity Matrices by Trees", J. Amer. Statist. Assoc., v.62, 1967, pp.1140-1158.

_____, "Direct Clustering of a Data Matrix", J. Amer. Statist. Assoc., v.67, 1972, pp.123-129.

_____, Clustering Algorithms, John Wiley, N.Y., 1975, 351p.

Jardine, N. and Sibson, R., "The Construction of Hierarchical and Non-hierarchical Classifications", Comp. J., v.11, 1968, pp. 177-184.

_____, "Choice of Methods for Automatic Classifications", Comp. J., v.14, 1971, pp.404-406.

Jarvis, R.A. and Patrick, E.A., "Clustering Using a Similarity Measure Based on Shared Near Neighbours", IEEE Trans. Comp., v.22, no.11, 1973, pp.1025-1034.

Jensen, R.E., "A Dynamic Programming Algorithm for Cluster Analysis", Op. Res., v.17, 1969, pp.1034-57.

Johnson, R.M., "Q-Analysis of Large Samples", Jour. Mark. Res., v. VII, 1970, pp.104-5.

Johnson, S.C., "Hierarchical Clustering Schemes", Psychometrika, v.32, no.3, 1967, pp.241-254.

Jones, K.S. and Jackson, D., "Current Approaches to Classification and Clump-finding at the Cambridge Language Research Unit", Comp. J., v.10, 1967, PP.29-37.

King, B.F., "Stepwise Clustering Prcedures", J. Amer. Statist. Assoc., v.62, 1967, pp.86-101.

Koontz, W.L. etal, "A Branch and Bound Clustering Algorithm", IEEE Tran. Comp., v.24, no. 9, 1975, pp.908-914.

Kruskal, Jr. J.B., "On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem", Proc. Amer. Math. Soc., no.7, 1956, pp.48-50.

Kruskal, J.B., "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis", Psychometrika, v.29, 1964, pp.1-28.

Lance, G.N. and Williams, W.T., "Computer Programs for Hierarchical Polythetic Classification('Similarity Analysis')", Comp. J., v.9, no.1, 1966, pp.60-64.

_____,"A General Theory of Classificatory Sorting Strategies I, Hierarchical Systems", Comp. J., v.9, 1967, pp.373-380.

_____, "A General Theory of Classificatory Sorting Strategies II, Clustering Systems", Comp. J., v.10, 1967, pp.271-277.

Ling, R.F., "On the Theory and Construction of K-Cluster", Comp. J., v.15, 1972, pp.326-332.

MacQueen, J.B., "Some Methods for Classification and Analysis of Multivariate Observations", Proc. Symp. Math. Statist. and Probability, 5th, Berkerly, v.1, 1967, pp.281-297.

McQuitty, L.L., "Hierarchical Linkage Analysis for the Isolation of Types", Educational and Psychological Measurement, v.20, 1960, pp.55-67.

_____, "Hierarchical Syndrome Analysis", Educ. and Psycho. Measure., v.20, 1960, pp.293-304.

_____, "Hierarchical Classification by Multiple Linkage", Educ. and Psycho. Measure., v.30, 1970, pp.3-19.

Mcrae, D.J., "MIKCA: A Fortran IV Iterative K-means Cluster Analysis Program", Behavioral Sci., v.16, no.4, 1971, pp.423-424.

Marriot, F.H.C., "Practical Problems in a Method of Cluster Analysis", Biometrics, v.27, no.3, 1971, pp.501-14.

Morrison, D.G., "Measurement Problems in Cluster Analysis", Management Sci., v.13, 1967, pp. B-775-780.

Parks, J.M., "Fortran IV Program for Q-Mode Cluster Analysis on Distance Function with Printed Dendogram", Comp. Contrib. 46, Stat. Geol. Survey, University of Kansas, Lawrence, Kansas, 1970, 36p.

Patterson, J.M. and Whitaker, R.A., "CGROUP: Hierarchical Grouping Analysis with Optimal Contigenity Constraint Program", UBC Computer Centre Program, UBC, Vancouver, 1973, 20p.

Rand, W.M., "Objective Criteria for the Evaluation of Clustering Methods", J. Amer. Statist. Assoc., v.66, 1971, pp.846-850.

Rohlf, F.J., "Adaptive Hierarchical Clustering Schemes", Syst. Zool., v.19, no.1, 1970, pp.58-83.

Sawrey, W.L. etal, "An Objective Method of Grouping Profiles by Distance Functions and its Relation to Factor Analysis", Educ. and Psycho. Measure., v.20, 1960, pp. 651-673.

Shepard, R.N., "Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function I and II", Psychometrika, v. 27, 1966, pp. 125-140, 219-246.

Shepherd, M.J. and Willmott, A.J., "Cluster Analysis on the Atlas Computer", Comp. J.,v.11, 1968, pp.57-62.

Sibson, R., "SLINK: An Optimally Efficient Algorithm for the Single Link Cluster Method", Comp. J., v.16, 1973, pp.30-34.

Sokal, R.R. and Michenener, C.D., "A Statistical Method for Evaluating Systematic Relationships", Univ. Kansas, Sci. Bull. 38, 1958, pp.1409-1438.

Sokal, R.R. and Sneath, P.H.A., Principles of Numerical Taxonomy , Freeman, San Francisco, 1963, 377p.

Tse, A., "Scheduling of Post Office Letter Box Collection Routes - A Case Study", Comm. 541 Project, UBC, Vancouver, 1975, 26p.

Ward, Jr. J.H., "Hierarchical Grouping to Op imise an Objective Function", J.Amer.Statist.Assoc., v.58, 1963, pp.236-244.

Ward, Jr. J.H. and Hook, M.E., "Application of an Hierarchical Grouping Procedure to a Problem of Grouping Profiles", Educ. and Psycho. Measurement, v.23, 1963, pp.69-83.

Wishart, D., "An Algorithm for Hierarchical Classifications", Biometrics, v.22, no.1, 1969, pp.165-170.

_____,"Fortran II Programs for 8 Methods of Cluster Analysis(CLUSTRAN I)", Comp. Contrib. 38, State Geol. Survey, Univ. of Kansas, Lawrence, 1969, 47p.

Wolfe, J.H., "Pattern Clustering by Multivariate Mixture Analysis", Multivariate Behavioral Res., v.5, no.3, 1970, pp.329-350.

Wright, W.E., "An Axiomatic Specification of Euclidean Analysis", Comp. J., v.17, 1974, pp.355-364.

Zahn, C.T., "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters", IEEE Trans. Comp., v.20, no.1, 1971, pp.68-86.

APPENDIX A

Listing of MATRIX: a Computer Program for Generating Symmetric Distance Matrix

```
C
C      THIS IS MATRIX: PROGRAM FOR CALCULATING DISTANCE MATRIX
C      FOR CLUSTERING PROGRAM INPUTS
C
      DIMENSION X(150),Y(150),DIST(150,150),ADIST(150,150)
      DIMENSION DMAT(7000)
      READ, N
      DO 10 I=1,N
      READ(5,100) X(I),Y(I)
      WRITE(6,101) I,X(I),Y(I)
10    CONTINUE
      DO 20 I1=1,N
      DO 30 I2=1,N
30    DIST(I1,I2)=((X(I2)-X(I1))**2)+((Y(I2)-Y(I1))**2)
20    CONTINUE
      AMAX=0.
      AMIN=999999.9
      DO 40 I3=1,N
      DO 50 I4=1,N
      IF(DIST(I3,I4).GT.AMAX) AMAX=DIST(I3,I4)
50    IF(DIST(I3,I4).LT.AMIN) AMIN=DIST(I3,I4)
40    CONTINUE
      DO 60 I5=1,N
      DO 70 I6=1,N
70    ADIST(I5,I6)=DIST(I5,I6)/AMAX
60    CONTINUE
      DO 80 I7=1,N
80    CONTINUE
      K=0
      DO 90 I8=2,N
      N2=I8-1
      DO 95 I9=1,N2
      K=K+1
95    DMAT(K)=ADIST(I8,I9)
90    CONTINUE
      COUNT=K/10.
      KOUNT=K/10
      K1=KOUNT
      IF(COUNT.GT.KOUNT) K1=KOUNT+1
      DO 96 II=1,K1
      KBEG=(II*10)-9
      KEND=II*10
      IF(II.EQ.K1) KEND=K
      WRITE(7,113) (DMAT(K3),K3=KBEG,KEND)
96    WRITE(6,113) (DMAT(K2),K2=KBEG,KEND)
100   FORMAT(5X,2F7.0)
101   FORMAT(I5,2F10.2)
113   FORMAT(10F7.4)
      STOP
      END
$SIG
```

APPENDIX B

Listing and Sample Outputs from HIER: a Computer Program for Six Hierarchical Clustering Techniques

```
C
C      PROGRAM DRIVER --- MAIN PROGRAM
      DIMENSION X(10000)
      LIMIT=10000
      CALL CNTRL(X,LIMIT)
      WRITE(6,100)
100   FORMAT('1',15X,'END OF OUTPUT')
      STOP
      END
C
      SUBROUTINE CNTRL(X,LIMIT)
C
C   THIS SUBROUTINE ALLOCATES STORAGE, READS INPUT AND CONTROLS
C   EXECUTION FOR A HIERARCHICAL CLUSTERING JOB BASED ON A PROVIDED
C   SIMILARITY MATRIX.
C
C-----------------------------------------------------------------------
C   INPUT SPECIFICATIONS
C
C   CARD 1  TITLE CARD
C   CARD 2 INFORMATION FOR SUBROUTINES CLSTR AND TREE
C     COLS  1- 3  NE=NUMBER OF ENTITIES (DATA UNITS OR VARIABLES) TO BE
C                    CLUSTERED
C     COLS  4- 5  ISIGN=OPTION FOR SIMILARITY FUNCTION
C                 ISIGN=+1, DISTANCE MEASURE
C                 ISIGN=-1, CORRELATION MEASURE
C     COLS  6- 7  NTSV=TAPE UNIT ON WHICH CLSTR RESULTS ARE SAVED
C                 NTSV=7, PUNCH RESULTS ON CARDS
C                 NTSV.LE.0, DO NOT SAVE RESULTS
C     COLS  8- 9  NTIN=UNIT FROM WHICH SIMILARITY MATRIX IS READ
C                 NTIN=5, CARD READER
C                 NTIN.NE.5, DISK OR TAPE
C     COLS 10-12  INOPT=INPUT OPTION FOR SIMILARITY MATRIX
C                 INOPT.LE.0, EACH RECORD IS ONE ROW OF A LOWER TRIANG-
C                             ULAR MATRIX
C                 INOPT.GT.0, THE LOWER TRIANGULAR MATRIX IS CONSIDERED
C                             TO BE STORED BY ROWS IN ONE LONG LINEAR
C                             ARRAY AND IS READ IN BLOCKS *INOPT* LONG.
C     COLS 13-14  KOUT=OUTPUT OPTION
C                 KOUT=+2, STANDARD OUTPUT
C                 KOUT=-2, STANDARD OUTPUT PLUS PUNCHED SEQUENCE LIST
C                          FROM SUBROUTINE *TREE*
C
C***ANY PREPOSITIONING OF THE I/O UNITS NTSV AND NTIN MUST BE
C   ACCOMPLISHED IN PROGRAM DRIVER OR THROUGH USE OF CONTROL CARDS.
C
C   CARD 3  INPUT FORMAT FOR SIMILARITY MATRIX (20A4 FORMAT)
C   CARD(S) 4  SIMILARITY MATRIX
C   CARD 5  END OF RECORD CARD (7/8/9)
```

```
**INCLUDE CARDS 4 AND 5 ONLY IF THE SIMILARITY MATRIX IS ON CARDS***

CARD(S) 6  LABEL CARDS FOR ENTITIES.  THERE ARE TWO OPTIONS
  1.  INCLUDE 1 CARD WITH THE 4-CHARACTERS *NOLB* IN COLUMNS 1-4.
      UNDER THIS OPTION LABELS ARE NOT PRINTED ON THE TREE OUTPUT.

  2.  INCLUDE NE CARDS, COLUMNS 1 TO 20 CONTAINING A LABEL FOR ONE
      ENTITY.  ORDER THE LABEL CARDS IN THE SAME SEQUENCE AS THE
      ENTITIES ARE REPRESENTED IN THE SIMILARITY MATRIX.
-------------------------------------------------------------------

DECK SETUP SPECIFICATIONS

THE USER PROVIDES PROGRAM DRIVER WHICH PERFORMS THE FOLLOWING TASKS.
  1.  ASSIGNS INPUT/OUTPUT UNITS
  2.  ESTABLISHES THE DIMENSION OF THE X ARRAY AND SETS THIS
      DIMENSION EQUAL TO LIMIT.
  3.  CALLS SUBROUTINE CNTRL.
THE FOLLOWING EXAMPLE WILL SUFFICE IN MOST CASES.

    PROGRAM DRIVER (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
   ATAPE7=PUNCH,TAPE1,TAPE2)
    DIMENSION X(7000)
    LIMIT=7000
    CALL CNTRL(X,LIMIT)
    END

A SECOND JOB DEPENDENT SEGMENT IS SUBROUTINE METHOD.  THE USER
SELECTS AMONG THE SEVERAL ALTERNATIVE VERSIONS OF THIS SUBROUTINE TO
IMPLEMENT THE DESIRED CLUSTERING TECHNIQUE.

THE SUBPROGRAMS CNTRL, CLSTR, MTXIN, LFIND AND TREE GO IN EVERY JOB.
-------------------------------------------------------------------

THE X ARRAY IS PARTITIONED FOR STORAGE AS FOLLOWS
STORAGE FOR ARRAYS NEEDED AT ALL STAGES OF THE JOB
X(N1) TO X(N2-1)  NE WORDS--STORAGE OF THE II ARRAY
X(N2) TO X(N3-1)  NE WORDS--STORAGE OF THE JJ ARRAY
X(N3) TO X(N4-1)  NE WORDS--STORAGE OF THE SS ARRAY
X(N4) TO X(N5-1)  NE WORDS--STORAGE OF THE IL ARRAY
X(N5) TO X(N6-1)  NE WORDS--STORAGE OF THE JL ARRAY
X(N6) TO X(N7-1)  NE WORDS--STORAGE OF THE NEXT ARRAY
STORAGE FOR ARRAYS NEEDED IN SUBROUTINE CLSTR
M1=N7
X(M1) TO X(M2-1)  (NE*(NE-1))/2 WORDS--STORAGE OF THE S ARRAY
X(M2) TO X(M3-1)  NE WORDS--STORAGE OF THE LAST ARRAY
X(M3) TO X(M4-1)  NE WORDS--STORAGE OF THE NEAR ARRAY
X(M4) TO X(M5-1)  NE WORDS--STORAGE OF THE SREF ARRAY
X(M5) TO X(M6-1)  NE WORDS--STORAGE OF THE LIST ARRAY
```

```
X(M6) TO X(M7-1)  NE WORDS--STORAGE OF THE A ARRAY
X(M7) TO X(M8)    NE WORDS--STORAGE OF THE B ARRAY
STORAGE FOR ARRAYS NEEDED IN SUBROUTINE TREE (OVERLAY ARRAYS NEEDED
IN SUBROUTINE CLSTR)
L1=N7
X(L1) TO X(L2-1)  25*NE WORDS--STORAGE OF THE A ARRAY
X(L2) TO X(L3-1)  5*NE WORDS--STORAGE OF THE LABEL ARRAY
X(L3) TO X(L4-1)  NE WORDS--STORAGE OF THE LCLNO ARRAY
X(L4) TO X(L5-1)  NE WORDS--STORAGE OF THE LINE ARRAY
X(L5) TO X(L6-1)  NE WORDS--STORAGE OF THE IS ARRAY
X(L6) TO X(L7)    NE WORDS--STORAGE OF THE LAST ARRAY
```

```
  INTEGER FIRST
  DIMENSION X(1),FMT(20),TITLE(20),EPS(25)
  DATA RLB/'NOLB'/
  READ(5,1000) TITLE
  READ(5,1100) NE,ISIGN,NTSV,NTIN,INOPT,KOUT
  WRITE(6,2500) TITLE
  WRITE(6,2200) NE,ISIGN,NTSV,NTIN,INOPT,KOUT
PARTITION THE STORAGE ARRAY
  N1=1
  N2=N1+NE
  N3=N2+NE
  N4=N3+NE
  N5=N4+NE
  N6=N5+NE
  N7=N6+NE
  M2=N7+(NE*(NE-1))/2
  M3=M2+NE
  M4=M3+NE
  M5=M4+NE
  M6=M5+NE
  M7=M6+NE
  M8=M7+NE-1
  L2=N7+25*NE
  L3=L2+5*NE
  L4=L3+NE
  L5=L4+NE
  L6=L5+NE
  L7=L6+NE-1
CHECK FOR SUFFICIENT STORAGE
  MAX=M8
  IF(L7.GT.MAX) MAX=L7
  WRITE(6,2300) MAX,LIMIT
  IF(MAX.GT.LIMIT) STOP
READ THE SIMILARITY MATRIX
  READ(5,1000) FMT
  WRITE(6,2100) FMT
  CALL MTXIN(X(N7),INOPT,NE,NTIN,FMT)
```

```
C  READY TO CLUSTER
60    CALL CLSTR(X(N1),X(N2),X(N3),X(N4),X(N5),X(N6),X(N7),X(M2),X(M3),
     AX(M4),X(M5),X(M6),X(M7),TITLE,NE,ISIGN,NTSV)
C  READ LABEL CARD(S)
      FIRST=L2
      LAST=L2+4
      READ(5,1000) (X(I),I=FIRST,LAST)
      IF(X(FIRST).EQ.RLB) GO TO 80
C  READ REMAINING LABELS
      DO 70 K=2,NE
      FIRST=LAST+1
      LAST=LAST+5
70    READ(5,1000) (X(I),I=FIRST,LAST)
C  DRAW THE TREE CORRESPONDING TO THE CLUSTERING
80    MERGES=NE-1
      CALL TREE(X(N1),X(N2),X(N3),X(N4),X(N5),X(N6),X(N7),X(L2),X(L3),
     AX(L4),X(L5),X(L6),EPS,TITLE,MERGES,1,6,1,KOUT,NE)
      RETURN
1000 FORMAT(20A4)
1100 FORMAT(I3,3I2,I3,I2,I3)
2100 FORMAT(7H FORMAT,20A4)
2200 FORMAT(5H NE =,I8,/,8H ISIGN =,I5,/,7H NTSV =,I6,/,7H NTIN =,I6,
    A/,8H INOPT =,I5,/,7H KOUT =,I6)
2300 FORMAT(19H REQUIRED STORAGE =,I5,6H WORDS,/,
    A       19H ALLOTTED STORAGE =,I5,6H WORDS,/)
2500 FORMAT('1',//,20A4,//)
      END
C
      SUBROUTINE CLSTR(II,JJ,SS,IL,JL,NEXT,S,LAST,NEAR,SREF,LIST,A,B,
     ATITLE,N,ISIGN,NT)
C  IN THIS VERSION THE LOWER TRIANGULAR PORTION OF THE SIMILARITY MATRIX
C  IS STORED BY ROWS IN THE ONE-DIMENSIONAL ARRAY S.
C
C  THE FOLLOWING VARIABLES ARE SPECIFIED IN THE CALLING PROGRAM AND
C  ARE PASSED THROUGH THE ARGUMENT LIST
C    N=NUMBER OF OBJECTS TO BE CLUSTERED
C    S(J)=J-TH ELEMENT IN LOWER TRIANGULAR SIMILARITY MATRIX
C    ISIGN=OPTION SPECIFYING TYPE OF SIMILARITY FUNCTION USED
C    ISIGN=+1=DISTANCE MEASURE (DECREASING FUNCTION OF SIMILARITY)
C    ISIGN=-1=CORRELATION MEASURE (INCREASING FUNCTION OF SIMILARITY)
C    NT=TAPE UNIT ON WHICH THE RESULTS ARE SAVED
C    NT.LE.0=DO NOT SAVE RESULTS ON TAPE
C    NT=7=SAVE RESULTS ON PUNCHED CARDS
C    TITLE=IDENTIFYING TITLE FOR THIS RUN
C
C  THE FOLLOWING VARIABLES REPRESENT THE OUTPUT OF THE PROGRAM AND ARE
C  PASSED BACK THROUGH THE ARGUMENT LIST.  THESE RESULTS ARE READY FOR
C  SUBROUTINE TREE.
C    K=STAGE OF CLUSTERING
C    II(K)=LOWER NUMBERED CLUSTER MERGED AT STAGE K
```

```
C     JJ(K)=UPPER NUMBERED CLUSTER MERGED AT STAGE K
C     SS(K)=VALUE OF SIMILARITY FUNCTION ASSOCIATED WITH MERGE AT STAGE K
C     IL(K)=PRECEDING STAGE AT WHICH II(K) WAS LAST IN A MERGE
C     JL(K)=PRECEDING STAGE AT WHICH JJ(K) WAS LAST IN A MERGE
C     NEXT(K)=NEXT STAGE AT WHICH II(K) IS IN A MERGE
C
C   IN ADDITION, THE FOLLOWING VARIABLES PLAY IMPORTANT ROLES IN THE PROGRAM
C     NEAR(I)=ID NUMBER OF EXTREME ELEMENT IN ROW I OF THE LOWER
C              TRIANGULAR SIMILARITY MATRIX.
C     SREF(I)=SIMILARITY MEASURE FOR THE PAIR (I,NEAR(I))
C     LIST(I)=I-TH CLUSTER ID NUMBER IN SEQUENTIAL LIST OF CURRENT CLUSTERS
C     NCL=NUMBER OF CLUSTERS AT CURRENT STAGE
C     LAST(I)=STAGE NUMBER AT WHICH CLUSTER I WAS LAST IN A MERGE
C     A=WORKING AREA FOR SUBROUTINE METHOD
C     R=WORKING AREA FOR SUBROUTINE METHOD
C
C   THIS SUBROUTINE USES FUNCTION LFIND(I,J) TO FIND THE ADDRESS IN S
C   FOR THE SIMILARITY MEASURE BETWEEN CLUSTERS I AND J
      DIMENSION S(1),II(1),JJ(1),SS(1),IL(1),JL(1),NEXT(1),NEAR(1),
     ASREF(1),LIST(1),LAST(1),A(1),B(1)
      DIMENSION TITLE(20)
C   INITIALIZE VARIABLES AND SET CONSTANTS
      NCL=N
      K=1
      SIGN=ISIGN
      BIG=SIGN*1.E50
      CALL METHOD(S,NEAR,SREF,LIST,A,B,SREFX,SIGN,N,NCL,LREF,NREF,1)
C   INITIALIZE ARRAYS
      DO 10 J=1,N
      LAST(J)=0
      NEXT(J)=0
      LIST(J)=J
      SREF(J)=BIG
10    CONTINUE
C   FIND EXTREME ENTRY IN EACH ROW
      L=0
      DO 30 I=2,N
      I1=I-1
      DO 30 J=1,I1
      L=L+1
C   IN EFFECT S(L)=S(I,J)
      IF(((S(L)-SREF(I))*SIGN).GT.0.) GO TO 30
      NEAR(I)=J
      SREF(I)=S(L)
30    CONTINUE
C   MAIN LOOP. FIND EXTREME VALUE IN SREF ARRAY
40    SREFX=BIG
      DO 50 I=2,NCL
      LISTI=LIST(I)
      IF(((SREF(LISTI)-SREFX)*SIGN).GT.0) GO TO 50
```

```
      IREF=I
      LREF=LISTI
      SREFX=SREF(LISTI)
50    CONTINUE
C   LREF IS THE ROW NUMBER CONTAINING THE EXTREME ENTRY IN THE S ARRAY.
C   IF THERE ARE TIES, THEN LREF IS THE HIGHEST NUMBERED ROW WITH THIS
C   EXTREME VALUE.  HENCE LREF.GT.NEAR(LREF).  IREF IDENTIFIES THE
C   PLACEMENT OF LREF IN THE LIST ARRAY.
      NREF=NEAR(LREF)
      CALL METHOD(S,NEAR,SREF,LIST,A,B,SREFX,SIGN,N,NCL,LREF,NREF,2)
C   GENERATE MERGE DATA NEEDED FOR SUBROUTINE TREE
      II(K)=NREF
      JJ(K)=LREF
      SS(K)=SREFX
      IL(K)=LAST(NREF)
      JL(K)=LAST(LREF)
      LAST(NREF)=K
      IF(IL(K).EQ.0) GO TO 60
      ILK=IL(K)
      NEXT(ILK)=K
60    IF(JL(K).EQ.0) GO TO 70
      JLK=JL(K)
      NEXT(JLK)=K
70    K=K+1
C   TERMINATE IF N-1 MERGES HAVE OCCURED
      IF(K.EQ.N) GO TO 140
C   UPDATE FOR THE NEXT CYCLE
      NCL=NCL-1
      IF(IREF.GT.NCL) GO TO 90
C   UPDATE LIST ARRAY BY REMOVING LREF AND PUSHING DOWN THE LIST
      DO 80 I=IREF,NCL
80    LIST(I)=LIST(I+1)
C   UPDATE FOR NEXT CYCLE
90    CALL METHOD(S,NEAR,SREF,LIST,A,B,SREFX,SIGN,N,NCL,LREF,NREF,3)
      GO TO 40
C   CLUSTERING FINISHED AND ALL ANCILLARY INFORMATION GENERATED.
C   SAVE RESULTS AS DESIRED.
140   K=K-1
160   IF(NT.LE.0) RETURN
      WRITE(NT,2300) TITLE
      DO 170 I=1,K
170   WRITE(NT,2200) I,II(I),JJ(I),SS(I),IL(I),JL(I),NEXT(I)
      RETURN
2200  FORMAT(3I10,E16.8,3I10)
2300  FORMAT(20A4)
      END
C
      FUNCTION LFIND(I,J)
C   IF THE LOWER TRIANGULAR PORTION OF A SYMMETRIC MATRIX IS STORED BY
C   ROWS IN A ONE-DIMENSIONAL ARRAY, THEN THE ELEMENT (I,J) IN THE FULL
```

```
C   MATRIX IS ELEMENT LFIND(I,J) IN THE LINEAR ARRAY
      IF(I.GT.J) GO TO 10
C   ROW J, COLUMN I
      LFIND=((J-1)*(J-2))/2+I
      RETURN
C   ROW I, COLUMN J
10    LFIND=((I-1)*(I-2))/2+J
      RETURN
      END
C
      SUBROUTINE TREE(I,J,S,IL,JL,NEXT,A,LABEL,LCLNO,LINE,IS,LAST,EPS,
     ATITLE,N,KBEG,NT,INTRV,IPRNT,MAXIN)
C
C   DATA INPUT THROUGH CALLING SEQUENCE
C
C   N=HIGHEST STAGE NUMBER IN THE CLUSTER MERGE DATA (MUST BE EXACT)
C   KBEG=STAGE NUMBER AT WHICH THE TREE BEGINS, DEFAULT VALUE 1
C   NT=TAPE NUMBER FOR PRINTED OUTPUT, DEFAULT VALUE = 6
C   INTRV-INTERVAL OPTION FOR SEGMENTATION
C   INTRV=1=DEFAULT VALUE. CONSTRUCT EPS BY DIVIDING THE RANGE OF S INTO
C           25 EQUAL SEGMENTS
C   INTRV=2=EPS IS PROVIDED AS PAST OF THE ARGUMENT LIST
C   INTRV=3=THE IS ARRAY IS ALREADY CONSTRUCTED AND EPS IS PROVIDED FOR INFO
C   IPRNT=PRINT OPTION FOR INPUT INFORMATION
C   IABS(IPRNT)=1. PRINT ONLY TITLE AND *IS* ARRAY
C   IABS(IPRNT).NE.1. IN ADDITION PRINT THE CLUSTER MERGE DATA
C   IPRNT.LE.0. IN ADDITION, PUNCH THE SEQUENCE IN WHICH THE ENTITIES
C                 APPEAR IN THE TREE (NEEDED FOR POST-ANALYSIS OF DATA
C                 UNIT CLUSTERING IN SUBROUTINE *POSTOJ*).
C   EPS(M)=RIGHT ENDPOINT FOR THE MIN INTERVAL USED FOR SEGMENTING S
C   LABEL(M,IJ)=WITH OF 5 WORDS IDENTIFYING THE IJTH OBJECT
C   TITLE=ARRAY OF 20 WORDS FOR IDENTIFYING THE RUN.
C   K=INDEX IDENTIFYING STAGE NUMBER IN THE CLUSTERING
C           KTH STAGE
C   J(K)=UPPER NUMBERED CLUSTER IDENTIFICATION NUMBER IN THE MERGE AT THE
C           KTH STAGE
C   S(K)=VALUE OF THE CRITERION FUNCTION FOR THE MERGE AT THE KTH STAGE
C   IS(K)=CATEGORIZED VALUE OF S= INTEGER IN RANGE 1 TO 25
C   IL(K)=STAGE NUMBER WHEN I(K) WAS LAST IN A MERGE (0 FOR FIRST MERGE FOR I(
C   JL(K)=STAGE NUMBER WHEN J(K) WAS LAST IN A MERGE (0 FOR FIRST MERGE FOR J(
C   NEXT(K)=STAGE NUMBER WHEN I(K) TO NEXT IN A MERGE
C   MAXIN=HIGHEST CLUSTER ID NUMBER IN THE CLUSTER MERGE DATA
C
C   OTHER VARIABLES USED IN THE PROGRAM
C
C   LINE(I)=LINE NUMBER IN THE PRINTOUT AT WHICH I(K) IS CARRIED (AFTER
C           MOST RECENT MERGE)
C   LCLNO(L)=THE CLUSTER NUMBER TO BE PRINTED ON LINE L AT THE LEFT OF THE TRE
C   A(M,L)=THE MTH SEGMENT (OF 25) IN THE LTH LINE OF THE PRINTOUT
C   LAST(L)=FARTHEST RIGHT SEGMENT IN LINE L WHICH IS NOT BLANK
```

```
      REAL*4 LABEL
      DIMENSION I(N),J(N),S(N),IS(N),IL(N),JL(N),NEXT(N),
     AA(25,MAXIN),LAST(MAXIN),LCLNO(MAXIN)
      DIMENSION LINE(MAXIN),LABEL(5,MAXIN)
      DIMENSION EPS(25),TITLE(20)
      DATA BAR1,BLINK,BARS,BLANK/4H---I,4H    I,4H----,+H     /
      DATA RLB/'NOLB'/
C   DEFAULT VALUES
      IF(KBEG.LT.1) KBEG=1
      IF(INTRV.LT.1.OR.INTRV.GT.3) INTRV=1
      IF(NT.LE.0) NT=6
C   INITIALIZE ARRAYS
      NOBJ=N+1
      DO 10 K=1,NOBJ
      LINE(K)=0
      LCLNO(K)=0
      LAST(K)=0
      DO 10 L=1,25
      A(L,K)=BLANK
10    CONTINUE
C   SEGMENT THE S ARRAY
      GO TO (20,40,120),INTRV
C   CONSTRUCT INTERVALS OF EQUAL LENGTH
20    RANGE=S(N)-S(KBEG)
      DELTA=RANGE/25.
      EPS(1)=S(KBEG)+DELTA
      DO 30 K=2,24
30    EPS(K)=EPS(K-1)+DELTA
      EPS(25)=S(N)
C   CONSTRUCT THE IS ARRAY
40    IF(EPS(1).GT.EPS(2)) GO TO 70
C   S INCREASES WITH DISSIMILARITY (AS DOES A DISTANCE)
      KK=1
      DO 60 K=1,N
50    IF(S(K).LE.EPS(KK)) GO TO 60
      IF(KK.EQ.25) GO TO 60
      KK=KK+1
      GO TO 50
60    IS(K)=KK
      GO TO 120
C   S DECREASES WITH DISSIMILARITY (AS DOES A CORRELATION)
70    KK=24
      KKK=25
      NN=N+1
      DO 90 K=1,N
      KCOMP=NN-K
80    IF(S(KCOMP).LT.EPS(KK)) GO TO 90
      KKK=KK
      KK=KK-1
      IF(KK.EQ.0) GO TO 100
```

```
      GO TO 80
90    IS(KCOMP)=KKK
100   DO 110 K=1,KCOMP
110   IS(K)=1
C   PRINT INPUT TO TREE
120   WRITE(NT,2000) TITLE
      WRITE(NT,2100) KBEG,N
      WRITE(NT,2200)
      WRITE(NT,2300)
      M=1
      WRITE(NT,2400) M,S(KBEG),EPS(M)
      DO 130 M=2,25
      MM=M-1
130   WRITE(NT,2400) M,EPS(MM),EPS(M)
      IF(IABS(IPRNT).EQ.1) GO TO 150
C   PRINT THE CLUSTER MERGE DATA
      WRITE(NT,2000)  TITLE
      WRITE(NT,2500)
      DO 140 K=KBEG,N
      WRITE(NT,2600) K,I(K),J(K),S(K),IS(K),IL(K),JL(K),NEXT(K)
140   CONTINUE
C   START TREE WITH THE MOST SIMILAR PAIR
150   K=KBEG
      LNO=0
C   MERGE CLUSTERS I(K) AND J(K)
160   IK=I(K)
      JK=J(K)
C   SET LINE NUMBERS FOR OUTPUT
      IF(IL(K).NE.0) GO TO 170
      LNO=LNO+1
      LINE(IK)=LNO
      LCLNO(LNO)=IK
170   IF(JL(K).NE.0) GO TO 180
      LNO=LNO+1
      LINE(JK)=LNO
      LCLNO(LNO)=JK
C   FILL IN THE PRINT LINES
180   ISK=IS(K)
      KT=0
      ITEM=IK
190   LITEM=LINE(ITEM)
      IF(ISK-LAST(LITEM)-1) 225,200,210
C   ADD ONLY ONE MORE SEGEMNT FOR LINE(ITEM)
200   A(ISK,LITEM)=BARI
      LAST(LITEM)=ISK
      GO TO 225
C   ADD MORE THAN ONE SEGMENT
210   LBEG=LAST(LITEM)+1
      LEND=ISK-1
      DO 220 L=LBEG,LEND
```

```
220   A(L,LITEM)=BARS
      GO TO 200
C  REPEAT FOR CLUSTER J(K)
225   KT=KT+1
      IF(KT.NE.1) GO TO 230
      ITEM=JK
      GO TO 190
C  TAKE CARE OF ANY LINES BETWEEN I(K) AND J(K)
230   LIK=LINE(IK)
      LJK=LINE(JK)
      IF(LIK.GT.LJK) GO TO 240
      LBOT=LJK
      LTOP=LIK
      GO TO 250
240   LBOT=LIK
      LTOP=LJK
250   IF(LBOT.EQ.(LTOP+1)) GO TO 270
C  MUST FILL IN SOME VERTICAL CONNECTIONS
      LBEG=LTOP+1
      LEND=LBOT-1
      DO 260 L=LBEG,LEND
      IF(A(ISK,L).EQ.BARI) GO TO 260
      A(ISK,L)=BLINK
      LAST(L)=ISK
260   CONTINUE
C  UPDATE LINE NUMBER FOR NEW CLUSTER
270   LINE(IK)=(LINE(IK)+LINE(JK))/2
C  MERGE COMPLETE.  FIND NEXT STAGE
      KLAST=K
      K=NEXT(K)
      IF(K.GT.N.OR.K.LT.KBEG) GO TO 400
      IF(IL(K).LE.0) GO TO 280
      IF(JL(K).LE.0) GO TO 290
      GO TO 300
280   IL(K)=-IL(K)
      GO TO 160
290   JL(K)=-JL(K)
      GO TO 160
C  THIS MERGE INVOLVES THAT EACH HAVE MOR THAN ONE MEMBER.
C  BACKTRACK TO THE ROOT OF THE TREE ALONG THE UNEXPLORED BRANCH.
300   IF(IL(K).EQ.KLAST) GO TO 310
C  GO DOWN IL(K) BRANCH.  SET JL(K) SO WE KNOW NOT TO GO DOWN THAT BRANCH AGA
      JL(K)=-JL(K)
      K=IL(K)
      GO TO 320
C  GO DOWN JL(K) BRANCH,  SET IL(K) SO WE KNOW NOT TO GO DOWN THAT BRANCH AGA
310   IL(K)=-IL(K)
      K=JL(K)
320   IF(K.LT.1.OR.K.GT.N) GO TO 600
C  TEST TO SEE IF THE END HAS BEEN REACHED.  IL(K)=JL(K) IFF BOTH ZERO.
```

```
      IF(IL(K)-JL(K)) 330,160,350
330   IF(IL(K).EQ.0) GO TO 360
340   K=IL(K)
      GO TO 320
350   IF(JL(K).EQ.0) GO TO 340
360   K=JL(K)
      GO TO 320
C  PRINT THE TREE
400   WRITE(NT,2000) TITLE
      IF(LABEL(1,1).EQ.RLB) GO TO 420
      WRITE(NT,3000) (K,K=1,25)
      DO 410 L=1,LNO
      LL=LCLNO(L)
410   WRITE(NT,3100) (LABEL(K,LL),K=1,5),LL,(A(K,L),K=1,25)
      GO TO 440
C  LEAVE LABEL SPACES BLANK
420   WRITE(NT,3010) (K,K=1,25)
      DO 430 L=1,LNO
      LL=LCLNO(L)
430   WRITE(NT,3210) LL,(A(K,L),K=1,25)
C  TREE COMPLETE
440   IF(IPRNT.GT.0) RETURN
C  PUNCH SEQUENCE LIST
      WRITE(7,3900) TITLE
      WRITE(7,4000) (LCLNO(L),L=1,LNO)
      RETURN
C  ERROR, PRINT AS MUCH OF THE TREE AS HAS BEEN CONSTRUCTED
600   WRITE(NT,6000) KLAST,K
      GO TO 400
2000  FORMAT(1H1,20X,20A4,//)
2100  FORMAT(65H THIS RUN DEPICTS THE PORTION OF THE TREE GENERATED BETW
     AEEN STAGE, I5,11H AND STAGE ,I5,19H OF THE CLUSTERING.,/)
2200  FORMAT(63H THE CRITERION VALUES ARE SEGMENTED INTO THE FOLLOWING C
     ALASSES.,/)
2300  FORMAT(6H CLASS,5X,11HLOWER BOUND,5X,11HUPPER BOUND,//)
2400  FORMAT(1X,I5,2E16.8)
2500  FORMAT(1H ,9X,1HK,9X,1H1,9X,1HJ,15X,1H5,8X,2HI5,8X,2HIL,8X,2HJL,6X
     A,4HNEXT,//)
2600  FORMAT(1X,3I10,E16.8,4I10)
3000  FORMAT(10H ITEM NAME,12X,5HID NO,2X,25I4,//)
C   IF LOCAL CONVENTIONS PERMIT, RECOMMEN THAT THE CARRIAGE CONTROL
C   CHARACTER IN FORMATS 3100 AND 3200 ALLOW 66 LINES OF PRINT PER PAGE.
C   THAT IS, THE MARGINS AT THE TOP AND BOTTOM OF THE PAGE ARE SUPPRESSED
C   AND PRINTING IS SINGLE SPACE.
3100  FORMAT(1H ,5A4,I6,2X,25A4)
3010  FORMAT(5X,5HID NO,2X,25I4,//)
3210  FORMAT(5X,I6,2X,25A4)
3900  FORMAT(20A4)
4000  FORMAT(20I4)
6000  FORMAT(37H ERROR. WHILE BACKTRACKING FROM KLAST,I6,27H K WAS FOUND
```

```
     A OUT OF RANGE.,/,1X,3HK =,I20)
      END
C
      SUBROUTINE MTXIN(X,IOPT,NE,NTIN,FMT)
C   THIS SUBROUTINE READS A LOWER TRIANGULAR MATRIX *X* REPRESENTING
C   ASSOCIATION AMONG *NE* ENTITIES.  THE MATRIX IS READ FROM UNIT *NTIN*
C   IN FORMAT *FMT*.  THE MODE OF INPUT FOR THE MATRIX IS DETERMINED BY
C   THE *IOPT* PARAMETER AS FOLLOWS.
C     IOPT.LE.0, MATRIX IS READ IN LOWER TRIANGULAR FORM BY ROWS, EACH
C                ROW BEING A NEW RECORD.
C     IOPT.GT.0, MATRIX IS READ IN CONSTANT LENGTH BLOCKS, EACH *IOPT*
C                WORDS LONG.
      DIMENSION FMT(20),X(1)
      INTEGER FIRST
      IF(IOPT.LE.0) GO TO 30
C      READ THE SIMILARITY MATRIX IN BLOCKS IOPT LONG
      FIRST=1
      LAST=IOPT
10    READ(NTIN,FMT,END=60) (X(I),I=FIRST,LAST)
C   USE THE END OF RECORD CARD TO SIGNIFY END OF THE SIMILARITY MATRIX
20    FIRST=FIRST+IOPT
      LAST=LAST+IOPT
      GO TO 10
C   READ THE SIMILARITY MATRIX AS ROWS OF A LOWER TRIANGULAR MATRIX,
C   EACH ROW A RECORD.
30    FIRST=1
      LAST=1
      DO 50 K=2,NE
      READ(NTIN,FMT,END=200) (X(I),I=FIRST,LAST)
40    FIRST=LAST+1
      LAST=LAST+K
50    CONTINUE
C   PASS THE END OF FILE
      READ(NTIN,FMT,END=60) Z
210   WRITE(6,2500)
      GO TO 999
60    RETURN
C   ERROR MESSAGES
200   WRITE(6,2400)
      GO TO 220
220   WRITE(6,2600)  K,FIRST,LAST,Z,(X(I),I=FIRST,LAST)
999   STOP
2400  FORMAT(36H EOF ENCOUNTERED WHEN NONE EXPECTED.)
2500  FORMAT(30H NO EOF WHEN ONE WAS EXPECTED.)
2600  FORMAT(1X,3I10,F10.7,/,(1X,12F10.7))
      END
$SIG
```

```
      SUBROUTINE METHOD(S,NEAR,SREF,LIST,A,B,SREFX,SIGN,N,NCL,LREF,NREF,
     AJOB)
C
C   HIERARCHICAL CLUSTERING BY SINGLE LINKAGE.  THE LAGORITHM IS DERIVED
C   FROM
C   JOHNSON, S.C., HIERARCHICAL CLUSTERING SCHEMES, PSYCHOMETRIKA,
C   VOLUME 32, NUMBER 3, SEPTEMBER 1967, PP 241-254.
C
      DIMENSION S(1),NEAR(1),SREF(1),LIST(1),A(1),B(1)
      GO TO (10,15,20),JOB
C   JOB=1. INITIALIZATION
10    WRITE(6,3000)
3000  FORMAT(26HOSINGLE LINKAGE CLUSTERING)
      BIG=SIGN*1.E50
      RETURN
C   JOB=2, DUMMY ENTRY.
15    RETURN
C   JOB=3, UPDATE FOR NEXT ROUND.
20    CONTINUE
      DO 50 J=1,NCL
C   UPDATE ENTRIES IN S ARRAY ASSOCIATED WITH NREF
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 50
C   RECALL THAT LREF HAS BEEN REMOVED FORM LIST SO I NEED NOT BE TESTED
C   FOR EQUALITY WITH LREF
      LL=LFIND(I,LREF)
      LN=LFIND(I,NREF)
      IF(((S(LL)-S(LN))*SIGN).GE.0.) GO TO 35
      S(LN)=S(LL)
      IF(I.GT.NREF) GO TO 30
C   I.LT.NREF
C   CHECK WHETHER S(LN) HAS A BETTER VALUE THAN SREF(NREF)
      IF(((S(LN)-SREF(NREF))*SIGN).GT.0.) GO TO 50
      NEAR(NREF)=I
      SREF(NREF)=S(LN)
      GO TO 50
30    IF(I.GT.LREF) GO TO 40
C   I.GT.NREF.AND.I.LT.LREF
C   CHECK WHETHER S(LN) HAS A BETTER VALUE THAN SREF(I)
      IF(((S(LN)-SREF(I))*SIGN).GE.0.) GO TO 50
      SREF(I)=S(LN)
      NEAR(I)=NREF
      GO TO 50
35    IF(I.LT.LREF) GO TO 50
C   I.GT.LREF
C   UPDATE NEAR ARRAY FOR THOSE ROWS WHOSE EXTREME ELEMENT WAS LREF
40    IF(NEAR(I).NE.LREF) GO TO 50
      NEAR(I)=NREF
      SREF(I)=S(LN)
50    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE METHOD(S,NEAR,SREF,LIST,A,B,SREFX,SIGN,N,NCL,LREF,NREF,
     AJOB)
C
C   HIERERCHICAL CLUSTERING BY COMPLETE LINKAGE.  THE ALGORITHM IS
C   DERIVED FROM
C   JOHNSON, S.C., HIERARCHICAL CLUSTERING SCHEMES, PSYCHOMETRIKA,
C   VOLUME 32, NUMBER 3, SEPTEMBER 1967, PP 241-254.
C
      DIMENSION S(1),NEAR(1),SREF(1),LIST(1),A(1),B(1)
      GO TO (10,15,20),JOB
C   JOB=1. INITIALIZATION
10    WRITE(6,2000)
2000  FORMAT(28HOCOMPLETE LINKAGE CLUSTERING)
      BIG=SIGN*1.E50
      RETURN
C   JOB=2, DUMMY ENTRY.
15    RETURN
C   JOB=3, UPDATE FOR NEXT ROUND.
20    DO 30 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 30
C   RECALL THAT LREF HAS BEEN REMOVED FROM LIST SO I NEED NOT BE
C   TESTED FOR EQUALITY WITH LREF.
      LL=LFIND(I,LREF)
      LN=LFIND(I,NREF)
      IF(((S(LL)-S(LN))*SIGN).LE.0) GO TO 30
      S(LN)=S(LL)
30    CONTINUE
C   UPDATE THE NEAR AND SREF ARRAYS.  IF THE EXTREME ELEMENT IN ROW I
C   WAS EITHER LREF OR NREF, THEN IT IS NECESSARY TO FIND A NEW EXTREME
C   ELEMENT.  ROWS PRIOR TO NREF NEED NOT BE CONSIDERED.
40    DO 50 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 55
50    CONTINUE
55    IF(J.EQ.1) GO TO 80
60    SREF(I)=BIG
      J1=J-1
      DO 70 L=1,J1
      LISTL=LIST(L)
      LL=LFIND(I,LISTL)
      IF(((S(LL)-SREF(I))*SIGN).GE.0.) GO TO 70
      NEAR(I)=LISTL
      SREF(I)=S(LL)
70    CONTINUE
80    J=J+1
      IF(J.GT.NCL) RETURN
      I=LIST(J)
      IF(NEAR(I).EQ.LREF.OR.NEAR(I).EQ.NREF) GO TO 60
      GO TO 80
      END
```

```
      SUBROUTINE METHOD(S,NEAR,SREF,LIST,NUMBR,SUM,SREFX,SIGN,N,NCL,
     ALREF,NREF,JOB)
C
C   HIERARCHICAL CLUSTERING BY MINIMIZING THE AVERAGE DISTANCE OR
C   MAXIMIZING THE AVERAGE CORRELATION BETWEEN THE MERGED GROUPS.
C
C   THE ALGORITHM IS DERIVED FROM THE *GROUP AVERAGE* METHOD DESCRIBED IN
C     LANCE, G.N. AND W. T. WILLIAMS, A GENERAL THEORY OF CLASSIFICATORY
C     SORTING STRATEGIES, 1. HIERARCHICAL SYSTEMS, THE COMPUTER JOURNAL,
C     VOLUME 9, NUMBER 4, FEBRUARY 1967, PP373-380.
C
      DIMENSION S(1),NEAR(1),SREF(1),LIST(1),NUMBR(1),SUM(1)
      GO TO (10,25,30),JOB
C   JOB=1, INITIALIZE.
C   NUMBR(I)=NUMBER OF ENTITIES CURRENTLY IN THE I-TH CLUSTER
10    WRITE(6,2000)
2000  FORMAT(42HOAVERAGE LINKAGE BETWEEN THE MERGED GROUPS)
      DO 20 J=1,N
20    NUMBR(J)=1
      BIG=SIGN*1.E50
      RETURN
C   JOB=2, DUMMY ENTRY.
25    RETURN
C   JOB=3, UPDATE FOR NEXT ROUND.
C   UPDATE THE NEW CLUSTER
30    NUMBR(NREF)=NUMBR(NREF)+NUMBR(LREF)
C   UPDATE ENTRIES IN THE REDUCED SIMILARITY MATRIX.  THE ENTRIES ARE
C   THE SUM TOTAL OF SIMILARITY VALUES ASSOCIATED WITH ALL
C   PAIRWISE LINKS BETWEEN THE ELEMENTS OF THE TWO CLUSTERS.
      DO 40 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 40
C   RECALL THAT LREF HAS BEEN REMOVED FROM LIST AND THEREFORE I NEED NOT
C   BE TESTED FOR EQUALITY WITH LREF.
      LL=LFIND(I,LREF)
      LN=LFIND(I,NREF)
      S(LN)=S(LN)+S(LL)
40    CONTINUE
C   UPDATE THE NEAR AND SREF ARRAYS.  IF THE EXTREME ELEMENT IN ROW I
C   WAS EITHER LREF OR NREF, THEN IT IS NECESSARY TO FIND A NEW WXTREME
C   ELEMENT.  ROWS PRIOR TO NREF NEED NOT BE CONSIDERED.
      DO 50 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 55
50    CONTINUE
55    IF(J.EQ.1) GO TO 80
60    SREF(I)=BIG
      J1=J-1
      DO 70 L=1,J1
      LISTL=LIST(L)
```

```
      LL=LFIND(I,LISTL)
      SREFX=S(LL)/(NUMBR(I)*NUMBR(LISTL))
      IF(((SREFX-SREF(I))*SIGN).GE.0.) GO TO 70
      NEAR(I)=LISTL
      SREF(I)=SREFX
70    CONTINUE
80    J=J+1
      IF(J.GT.NCL) RETURN
      I=LIST(J)
      IF(NEAR(I).EQ.LREF.OR.NEAR(I).EQ.NREF) GO TO 60
      GO TO 80
      END
```

```
      SUBROUTINE METHOD(S,NEAR,SREF,LIST,NUMBR,SUM,SREFX,SIGN,N,NCL,
     ALREF,NREF,JOB)
C
C   HIERARCHICAL CLUSTERING BY MINIMIZING THE AVERAGE DISTANCE OR
C   MAXIMIZING THE AVERAGE CORRELATION WITHIN THE NEW GROUP.  THAT IS,
C   FOR EACH POTENTIAL MERGE THE AVERAGE OF ALL LINKAGES WITHIN THE
C   NEW GROUP IS CALCULATED.
      DIMENSION S(1),NEAR(1),SREF(1),LIST(1),NUMBR(1),SUM(1)
      GO TO (10,25,30),JOB
C       JOB=1, INITIALIZE.
C   NUMBR(I)=NUMBER OF ENTITIES CURRENTLY IN THE I-TH CLUSTER
C   SUM(I)=SUM OF ALL PAIRWISE SIMILARITIES AMONG ENTITIES IN THE I-TH
C          CLUSTER
10    WRITE(6,2000)
2000  FORMAT(37HOAVERAGE LINKAGE WITHIN THE NEW GROUP)
      DO 20 J=1,N
      NUMBR(J)=1
20    SUM(J)=0.
      BIG=SIGN*1.E50
      RETURN
C   JOB=2, DUMMY ENTRY.
25    RETURN
C   JOB=3, UPDATE FOR NEXT ROUND.
C   UPDATE THE NEW CLUSTER
30    NUMBR(NREF)=NUMBR(NREF)+NUMBR(LREF)
      LN=LFIND(LREF,NREF)
      SUM(NREF)=SUM(NREF)+SUM(LREF)+S(LN)
C   UPDATE ENTRIES IN THE REDUCED SIMILARITY MATRIX.  THE ENTRIES ARE
C   THE SUM TOTAL OF SIMILARITY VALUES ASSOCIATED WITH ALL
C   PAIRWISE LINKS BETWEEN THE ELEMENTS OF THE TWO CLUSTERS.
      DO 40 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 40
C   RECALL THAT LREF HAS BEEN REMOVED FROM LIST AND THEREFORE I NEED NOT
C   BE TESTED FOR EQUALITY WITH LREF.
      LL=LFIND(I,LREF)
      LN=LFIND(I,NREF)
      S(LN)=S(LN)+S(LL)
40    CONTINUE
C   UPDATE THE NEAR AND SREF ARRAYS.  IF THE EXTREME ELEMENT IN ROW I
C   WAS EITHER LREF OR NREF, THEN IT IS NECESSARY TO FIND A NEW EXTREME
C   ELEMENT.  ROWS PRIOR TO NREF NEED NOT BE CONSIDERED.
      DO 50 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 55
50    CONTINUE
55    IF(J.EQ.1) GO TO 80
60    SREF(I)=BIG
      J1=J-1
      DO 70 L=1,J1
```

```
      LISTL=LIST(L)
      LL=LFIND(I,LISTL)
      NTOT=NUMBR(I)+NUMBR(LISTL)
      NTOT=((NTOT)*(NTOT-1))/2
      SREFX=(SUM(I)+SUM(LISTL)+S(LL))/NTOT
      IF(((SREFX-SREF(I))*SIGN).GE.0.) GO TO 70
      NEAR(I)=LISTL
      SREF(I)=SREFX
70    CONTINUE
80    J=J+1
      IF(J.GT.NCL) RETURN
      I=LIST(J)
      IF(NEAR(I).EQ.LREF.OR.NEAR(I).EQ.NREF) GO TO 60
      GO TO 80
      END
```

```
      SUBROUTINE METHOD(S,NEAR,SREF,LIST,NUMBR,SUM,SREFX,SIGN,N,NCL,
     ALREF,NREF,JOB)
C
C   HIERARCHICAL CLUSTERING BY CENTROID SORTING
C
C   THE PARTICULAR ALGORITHM USED HERE IS DESCRIBED IN
C     LANCE, G.N. AND W.T. WILLIAMS, A GENERAL THEORY OF CLASSIFICATORY
C     SORTING STRATEGIES, 1. HIERARCHICAL SYSTEMS, THE COMPUTER JOURNAL,
C     VOLUME 9, NUMBER 4, FEBRUARY 1967, PP373-380.
      DIMENSION S(1),NEAR(1),SREF(1),LIST(1),NUMBR(1),SUM(1)
      GO TO (10,25,30),JOB
C   JOB=1, INITIALIZE.
C   NUMBR(I)=NUMBER OF ENTITIES CURRENTLY IN THE I-TH CLUSTER
C         CLUSTER
10    WRITE(6,2000)
2000  FORMAT(42HOCENTROID CLUSTERING.  BEWARE OF REVERSALS)
      DO 20 J=1,N
20    NUMBR(J)=1
      BIG=SIGN*1.E50
      RETURN
C   JOB=2, DUMMY ENTRY.
25    RETURN
C   JOB=3, UPDATE FOR NEXT ROUND.
C   UPDATE THE NEWCLUSTER
30    NTOT=NUMBR(NREF)+NUMBR(LREF)
      TOT=NTOT
      ALL=NUMBR(LREF)/TOT
      ALN=NUMBR(LREF)/TOT
      PROD=ALN*ALL
      LBET=LFIND(LREF,NREF)
      DO 40 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 40
C   RECALL THAT LREF HAS BEEN REMOVED FROM LIST AND TEREFORE I NEED NOT
C   BE TESTED FOR EQUALITY WITH LREF.
      LL=LFIND(I,LREF)
      LN=LFIND(I,NREF)
      S(LN)=ALL*S(LL)+ALN*S(LN)-PROD*S(LBET)
40    CONTINUE
C   UPDATE THE NEAR AND SREF ARRAYS.  IF THE EXTREME ELEMENT IN ROW I
C   WAS EITHER LREF OR NREF, THEN IT IS NECESSARY TO FIND A NEW EXTREME
C   ELEMENT.  ROWS PRIOR TO NREF NEED NOT BE CONSIDERED.
      DO 50 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 55
50    CONTINUE
55    IF(J.EQ.1) GO TO 80
60    SREF(I)=BIG
      J1=J-1
      DO 70 L=1,J1
```

```
      LISTL=LIST(L)
      LL=LFIND(I,LISTL)
      IF(((S(LL)-SREF(I))*SIGN).GE.0.) GO TO 70
      NEAR(I)=LISTL
      SREF(I)=S(LL)
70    CONTINUE
80    J=J+1
      IF(J.GT.NCL) RETURN
      I=LIST(J)
      IF(NEAR(I).EQ.LREF.OR.NEAR(I).EQ.NREF) GO TO 60
      GO TO 80
      END
$SIG
```

```
      SUBROUTINE METHOD(S,NEAR,SREF,LIST,A,B,SREFX,SIGN,N,NCL,LREF,NREF,
     AJOB)
C
C     HIERARCHICAL CLUSTERING BY THE MIDIAN METHOD OF
C       GOWER, J.C., A COMPARISON OF SOME METHODS OF CLUSTER ANALYSIS,
C       BIOMETRICS, VOLUME 23, NUMBER 4, DECEMBER 1967, PP 623-637.
C
C
      DIMENSION S(1),NEAR(1),SREF(1),LIST(1),A(1),B(1)
      GO TO (10,15,20),JOB
C     JOB=1. INITIALIZATION
10    WRITE(6,2000)
2000  FORMAT(44HOMEDIAN METHOD OF GOWER, BEWARE OF REVERSALS)
      BIG=SIGN*1.E50
      RETURN
C     JOB=2, DUMMY ENTRY.
15    RETURN
C     JOB=3, UPDATE FOR NEXT ROUND.
20    LBET=LFIND(LREF,NREF)
      DO 30 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 30
C     RECALL THAT LREF HAS BEEN REMOVED FROM LIST SO I NEED NOT BE
C     TESTED FOR EQUALITY WITH LREF.
      LL=LFIND(I,LREF)
      LN=LFIND(I,NREF)
C     IF S IS A DECREASING FUNCTION OF SIMILARITY (E.G. DISTANCE) THEN
C        S(LN)=(S(LN)+S(LL))/2,-S(LBET)/4.
C     IF S IS AN INCREASING FUNCTION OF SIMILARITY (E.G. CORRELATION) THEN
C        S(LN)=(S(LN)+S(LL))/2.+(1.-S(LBET))/4.
      S(LN)=(S(LN)+S(LL))/2.-S(LBET)/4.
30    CONTINUE
C     UPDATE THE NEAR AND SREF ARRAYS.  IF THE EXTREME ELEMENT IN ROW I
C     WAS EITHER LREF OR NREF. THEN IT IS NECESSARY TO FIND A NEW WXTREME
C     ELEMENT.  ROWS PRIOR TO NREF NEED NOT BE CONSIDERED.
40    DO 50 J=1,NCL
      I=LIST(J)
      IF(I.EQ.NREF) GO TO 55
50    CONTINUE
55    IF(J.EQ.1) GO TO 80
60    SREF(I)=BIG
      J1=J-1
      DO 70 L=1,J1
      LISTL=LIST(L)
      LL=LFIND(I,LISTL)
      IF(((S(LL)-SREF(I))*SIGN).GE.0.) GO TO 70
      NEAR(I)=LISTL
      SREF(I)=S(LL)
70    CONTINUE
80    J=J+1
      IF(J.GT.NCL) RETURN
      I=LIST(J)
      IF(NEAR(I).EQ.LREF.OR.NEAR(I).EQ.NREF) GO TO 60
      GO TO 80
      END
```

CLUSTER TRIAL RUN -1

```
NE =      80
ISIGN =    1
NTSV =    -1
NTIN =     7
INOPT =   10
KOUT =     2
REQUIRED STORAGE = 4120 WORDS
ALLOTTED STORAGE = 7000 WORDS

FORMAT(10F7.4)

AVERAGE LINKAGE WITHIN THE NEW GROUP
```

THIS RUN DEPICTS THE PORTION OF THE TREE GENERATED BETWEEN STAGE 1 AND STAGE 79 OF THE CLUSTERING.

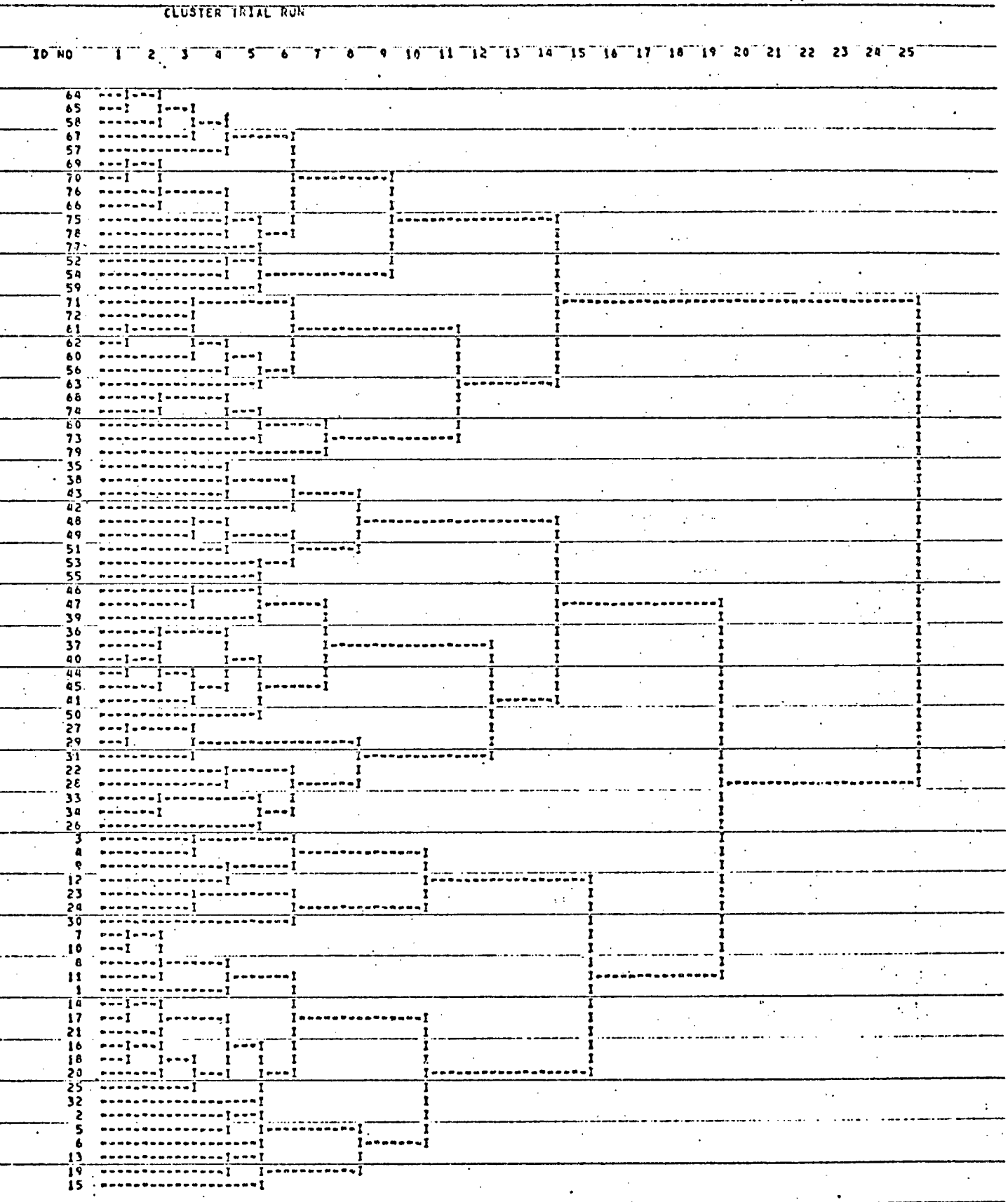
THE CRITERION VALUES ARE SEGMENTED INTO THE FOLLOWING CLASSES.

CLASS	LOWER BOUND	UPPER BOUND
1	0.21200000E-01	0.35035290E-01
2	0.35035290E-01	0.48870590E-01
3	0.48870590E-01	0.62705870E-01
4	0.62705870E-01	0.76541120E-01
5	0.76541120E-01	0.90376370E-01
6	0.90376370E-01	0.10421160E 00
7	0.10421160E 00	0.11804680E 00
8	0.11804680E 00	0.13188210E 00
9	0.13188210E 00	0.14571730E 00
10	0.14571730E 00	0.15955260E 00
11	0.15955260E 00	0.17338780E 00
12	0.17338780E 00	0.18722310E 00
13	0.18722310E 00	0.20105830E 00
14	0.20105830E 00	0.21489360E 00
15	0.21489360E 00	0.22872880E 00
16	0.22872880E 00	0.24256410E 00
17	0.24256410E 00	0.25639930E 00
18	0.25639930E 00	0.27023460E 00
19	0.27023460E 00	0.28406980E 00
20	0.28406980E 00	0.29790510E 00
21	0.29790510E 00	0.31174030E 00
22	0.31174030E 00	0.32557560E 00
23	0.32557560E 00	0.33941090E 00
24	0.33941090E 00	0.35324610E 00
25	0.35324610E 00	0.36708240E 00

CLUSTER TRIAL RUN

K	I	J	S	IS	IL	JL	NEXT
1	64	65	0,21200000E-01	1	0	0	12
2	40	44	0,21200000E-01	1	0	0	9
3	7	10	0,21200000E-01	1	0	0	10
4	69	70	0,26500000E-01	1	0	0	13
5	14	17	0,26500000E-01	1	0	0	14
6	61	62	0,31700000E-01	1	0	0	20
7	27	29	0,31700000E-01	1	0	0	23
8	16	18	0,31700000E-01	1	0	0	15
9	40	45	0,35266650E-01	2	2	0	24
10	7	8	0,35266650E-01	2	3	0	19
11	33	34	0,37000000E-01	2	0	0	50
12	58	64	0,38850000E-01	2	0	1	21
13	69	76	0,42333320E-01	2	4	0	16
14	14	21	0,42333320E-01	2	5	0	33
15	16	20	0,45833300E-01	2	8	0	22
16	66	69	0,46733310E-01	2	0	13	30
17	68	74	0,47600000E-01	2	0	0	39
18	36	37	0,47600000E-01	2	0	0	41
19	7	11	0,48483330E-01	2	10	0	34
20	60	61	0,49366630E-01	3	0	6	35
21	58	67	0,53783320E-01	3	12	0	36
22	16	25	0,54666630E-01	3	15	0	33
23	27	31	0,56433320E-01	3	7	0	68
24	40	41	0,57333320E-01	3	9	0	41
25	71	72	0,58200000E-01	3	0	0	62
26	48	49	0,58200000E-01	3	0	0	40
27	46	47	0,58200000E-01	3	0	0	51
28	23	24	0,58200000E-01	3	0	0	58
29	3	4	0,58200000E-01	3	0	0	57
30	66	75	0,63499970E-01	4	16	0	45
31	52	54	0,63499980E-01	4	0	0	56
32	2	5	0,63499980E-01	4	0	0	47
33	14	16	0,64495200E-01	4	14	22	49
34	1	7	0,64549980E-01	4	0	19	64
35	56	60	0,65249970E-01	4	0	20	54
36	57	58	0,68779940E-01	4	0	21	63
37	35	38	0,68799970E-01	4	0	0	42
38	13	19	0,68799970E-01	4	0	0	52
39	68	80	0,70533330E-01	4	17	0	46
40	48	51	0,74066630E-01	4	26	0	61
41	36	40	0,74079930E-01	4	18	24	55
42	35	43	0,74099950E-01	4	37	0	59
43	22	28	0,74100010E-01	4	0	0	60
44	9	12	0,74100010E-01	4	0	0	57
45	66	78	0,75479920E-01	4	30	0	53
46	68	73	0,77600000E-01	5	39	0	65
47	2	6	0,77600000E-01	5	32	0	67
48	53	55	0,79400000E-01	5	0	0	61
49	14	32	0,79928510E-01	5	33	0	64
50	26	33	0,81099980E-01	5	0	11	60
51	39	46	0,84633290E-01	5	0	27	66
52	13	15	0,84666600E-01	5	38	0	67
53	66	77	0,85661820E-01	5	45	0	63
54	56	63	0,86769930E-01	5	35	0	62
55	36	50	0,87186650E-01	5	41	0	66
56	52	59	0,88199970E-01	5	31	0	70
57	3	9	0,90849990E-01	6	29	44	71
58	23	30	0,91699950E-01	6	20	0	71
59	35	42	0,94349980E-01	6	42	0	69
60	22	26	0,96299990E-01	6	43	50	68
61	48	53	0,96299980E-01	6	40	48	69
62	56	71	0,10128540E 00	6	54	25	73
63	57	66	0,10173320E 00	6	36	53	70
64	1	14	0,10229470E 00	6	34	49	72
65	68	79	0,10687990E 00	7	46	0	73
66	36	39	0,11311540E 00	7	55	51	74
67	2	13	0,11816650E 00	8	47	52	72
68	22	27	0,11885700E 00	8	60	23	74
69	35	48	0,12610540E 00	8	59	61	76
70	52	57	0,13414180E 00	9	56	63	75
71	3	23	0,14916660E 00	10	57	58	77
72	1	2	0,15074950E 00	10	64	67	77
73	56	68	0,16033620E 00	11	62	65	75
74	22	36	0,18145320E 00	12	68	66	76
75	52	56	0,20353080E 00	14	70	73	79
76	22	35	0,20500960E 00	14	74	69	78
77	1	3	0,21781300E 00	15	72	71	78
78	1	22	0,28134730E 00	19	77	76	79
79	1	52	0,36708240E 00	25	78	75	0

CLUSTER TRIAL RUN



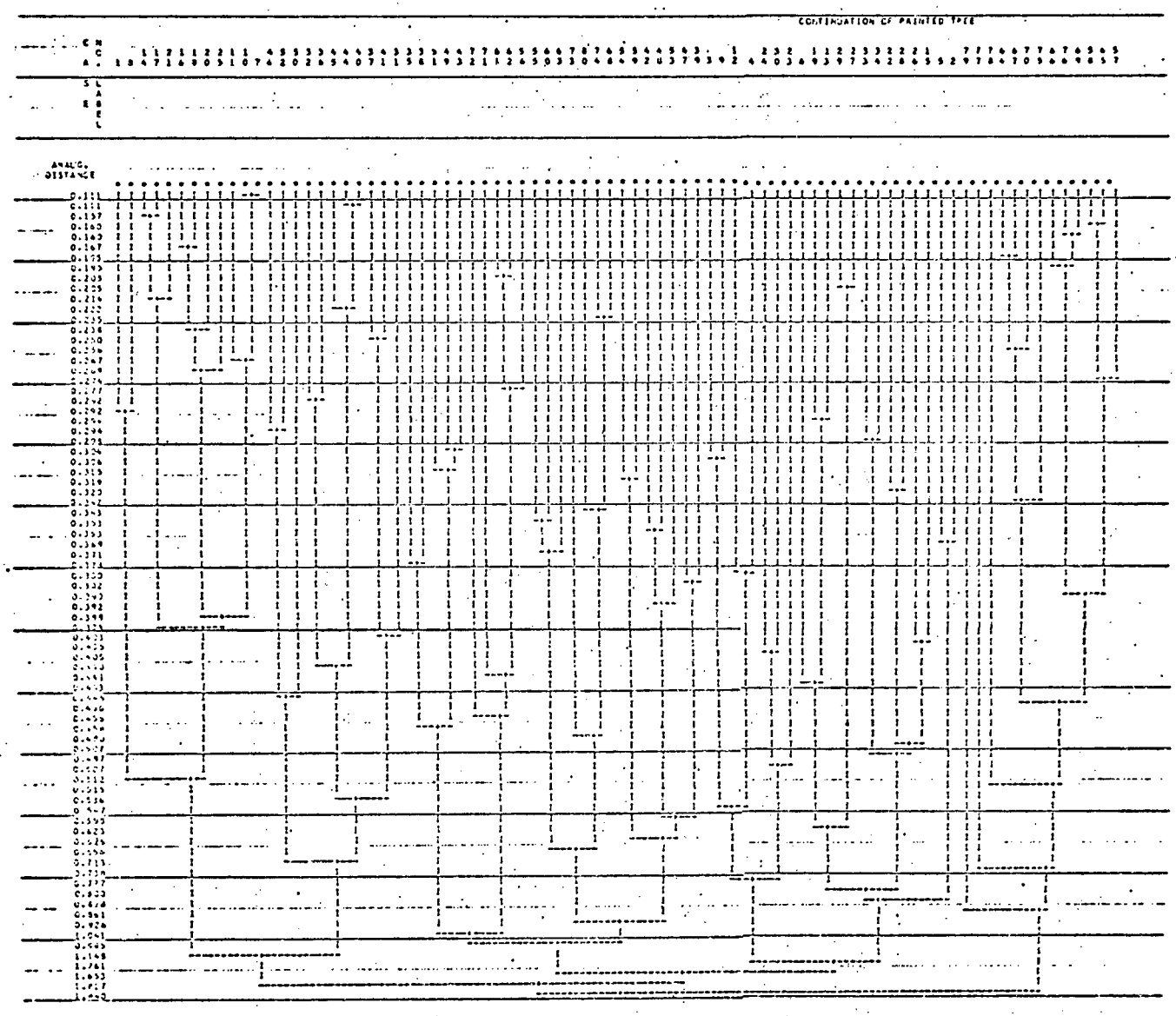
APPENDIX C

Sample Outputs from Program UBC:BMDP2M

```
BMDP2M - CLUSTER ANALYSIS OF CASES                    PROGRAM REVISED FEBRUARY 26, 1973
HEALTH SCIENCES COMPUTING FACILITY                    WRITEUP REVISED SEPTEMBER, 1971
UNIVERSITY OF CALIFORNIA, LOS ANGELES


PROBLEM CONTROL CARDS
        PROB TITLE IS 'CLUSTER TRIAL RUN - TDATA1 - 2M'./
        INPUT VARIABLE=2.
              CASE=80.
              FORMAT='(5X,2F7.0)'./
        PROC SUMOFSQ. STAND./
        PRINT DATA. DISTANCE. VERTICAL./
        END/


 PROBLEM TITLE . . . . . .CLUSTER TRIAL RUN - TDATA1 - 2M

 NUMBER OF VARIABLES TO READ IN. . . . . .        2
 NUMBER OF VARIABLES ADDED BY TRANSFORMATIONS. .  0
 TOTAL NUMBER OF VARIABLES . . . . . .        2
 NUMBER OF CASES TO READ IN. . . . . .       80
 CASE LABELING VARIABLES . . . . . .  0   0
 LIMITS AND MISSING VALUE CHECKED BEFORE TRANSFORMATIONS
 INPUT TAPE NUMBER . . . . . .        5
 REWIND INPUT TAPE PRIOR TO READING DATA . . . .     NO
 INPUT FORMAT. . . . . .(5X,2F7.0)
 PRINT DISTANCE MATRIX . . . . . .     YES
 TYPE OF TREE PRINTED. . . . . .VERTICAL
 CALCULATING PROCEDURE . . . . . . SUM-SQR
 STANDARDIZATION ON INPUT DATA . . . . . .     YES
 PRINT INPUT DATA MATRIX AFTER STANDARDIZATION .     YES
```

NO.	NAME	STANDARDIZED INPUT DATA				
1		-1.597	-0.879			
2		-1.597	0.361			
3		-1.597	1.104			
4		-1.597	1.649			
5		-1.471	0.708			
6		-1.471	-0.135	54	0.666	-0.086
7		-1.346	-1.126	55	0.666	0.856
8		-1.346	-0.730	56	0.666	1.600
9		-1.295	1.055	57	0.791	-1.126
10		-1.295	-1.027	58	0.917	-0.578
11		-1.220	-1.325	59	0.867	0.162
12		-1.220	1.600	60	0.917	1.104
13		-1.170	-0.234	61	0.917	1.699
14		-1.094	-0.879	62	0.967	1.501
15		-1.094	0.410	63	1.043	0.708
16		-1.044	-1.275	64	1.043	-0.680
17		-1.044	-0.730	65	1.043	-0.879
18		-0.969	-1.126	66	1.168	-1.126
19		-0.893	-0.135	67	1.168	-0.532
20		-0.843	-1.374	68	1.219	0.460
21		-0.868	-0.879	69	1.294	-1.027
22		-0.843	0.708	70	1.294	-0.779
23		-0.843	1.352	71	1.244	1.798
24		-0.843	1.897	72	1.294	1.352
25		-0.717	-1.126	73	1.420	0.856
26		-0.717	0.261	74	1.344	0.261
27		-0.591	-0.383	75	1.420	-0.433
28		-0.591	0.906	76	1.420	-1.126
29		-0.541	-0.185	77	1.546	-1.523
30		-0.466	1.749	78	1.671	-0.879
31		-0.466	-0.730	79	1.747	-0.234
32		-0.466	-1.275	80	1.621	0.410
33		-0.315	0.311			
34		-0.340	0.608			
35		-0.290	1.302			
36		-0.214	-1.126			
37		-0.164	-0.779			
38		-0.139	0.955			
39		-0.089	-0.135			
40		0.037	-1.374			
41		0.037	-0.631			
42		0.037	0.460			
43		0.037	1.352			
44		0.087	-1.275			
45		0.163	-1.126			
46		0.288	-0.383			
47		0.238	0.063			
48		0.288	0.708			
49		0.213	1.104			
50		0.414	-0.879			
51		0.414	1.352			
52		0.565	-0.482			
53		0.515	0.410			

ANALG.	DIST.	VALUES OF VARIABLES OF CLUSTERS		
1	0.111	-1.321	-1.077	2.000
2	0.111	0.062	-1.325	2.000
3	0.157	-1.069	-0.804	2.000
4	0.160	0.980	-0.928	2.000
5	0.160	1.357	-1.077	2.000
6	0.167	-1.006	-1.201	2.000
7	0.195	1.106	-0.606	2.000
8	0.195	1.294	-1.093	3.000
9	0.205	0.942	1.600	2.000
10	0.205	-0.566	-0.264	2.000
11	0.214	-1.002	-0.829	3.000
12	0.222	0.096	-1.259	3.000
13	0.235	1.282	0.361	2.000
14	0.238	-0.952	-1.259	3.000
15	0.250	-0.063	-0.705	2.000
16	0.256	1.168	-0.664	3.000
17	0.267	-1.287	-1.160	3.000
18	0.269	-0.893	-1.226	4.000
19	0.274	0.517	-0.994	3.000
20	0.277	0.850	1.600	3.000
21	0.292	-0.340	-1.201	2.000
22	0.292	-1.471	-0.804	2.000
23	0.294	-1.031	-0.185	2.000
24	0.294	0.427	-0.433	2.000
25	0.298	-0.327	0.460	2.000
26	0.304	0.125	1.228	2.000
27	0.306	-1.446	1.079	2.000
28	0.315	0.221	1.249	3.000
29	0.319	0.766	0.038	2.000
30	0.320	-0.717	0.807	2.000
31	0.342	1.231	-0.606	4.000
32	0.343	1.395	0.377	3.000
33	0.353	0.791	0.980	2.000
34	0.353	0.163	0.584	2.000
35	0.369	-1.534	0.534	2.000
36	0.371	0.875	0.889	3.000
37	0.378	-0.214	1.129	2.000
38	0.380	-1.409	1.625	2.000
39	0.382	0.075	-0.036	2.000
40	0.390	1.106	-1.044	6.000
41	0.392	0.280	0.526	3.000
42	0.399	-1.062	-1.197	7.000
43	0.373	-1.044	-1.087	10.000
44	0.403	-0.198	-0.713	3.000
45	0.405	-0.906	0.336	2.000
46	0.405	-0.654	1.823	2.000
47	0.440	-0.079	-1.236	5.000
48	0.441	0.948	1.649	4.000
49	0.443	-1.178	-0.168	3.000
50	0.446	0.423	-0.561	3.000
51	0.456	1.156	-0.869	10.000
52	0.456	1.018	1.590	5.000
53	0.458	0.047	1.213	5.000
54	0.480	1.401	0.497	4.000
55	0.507	-0.811	0.571	4.000
56	0.497	-0.650	0.534	6.000
57	0.507	-0.717	1.666	3.000
58	0.512	-1.115	-1.040	12.000
59	0.515	1.203	-0.870	11.000
60	0.536	-0.123	-1.040	8.000
61	0.547	-1.427	1.352	4.000
62	0.598	0.198	0.301	5.000
63	0.623	-0.933	-0.214	5.000
64	0.626	0.360	0.226	7.000
65	0.656	1.176	0.265	7.000
66	0.713	0.026	-0.915	11.000
67	0.738	1.231	-0.924	12.000
68	0.777	-1.123	1.487	7.000
69	0.800	-0.779	0.194	11.000
70	0.828	-0.895	0.246	13.000
71	0.861	1.271	-0.871	13.000
72	0.926	0.768	0.446	14.000
73	1.041	0.532	1.402	10.000
74	0.985	0.670	0.844	24.000
75	1.148	-0.570	-0.980	23.000
76	1.261	-0.975	0.680	20.000
77	1.653	-0.078	0.770	44.000
78	1.817	-0.247	0.169	67.000
79	1.840	0.000	0.000	80.000


CONTINUATION OF PRINTED TREE
ANALYSIS DISTANCE

APPENDIX D

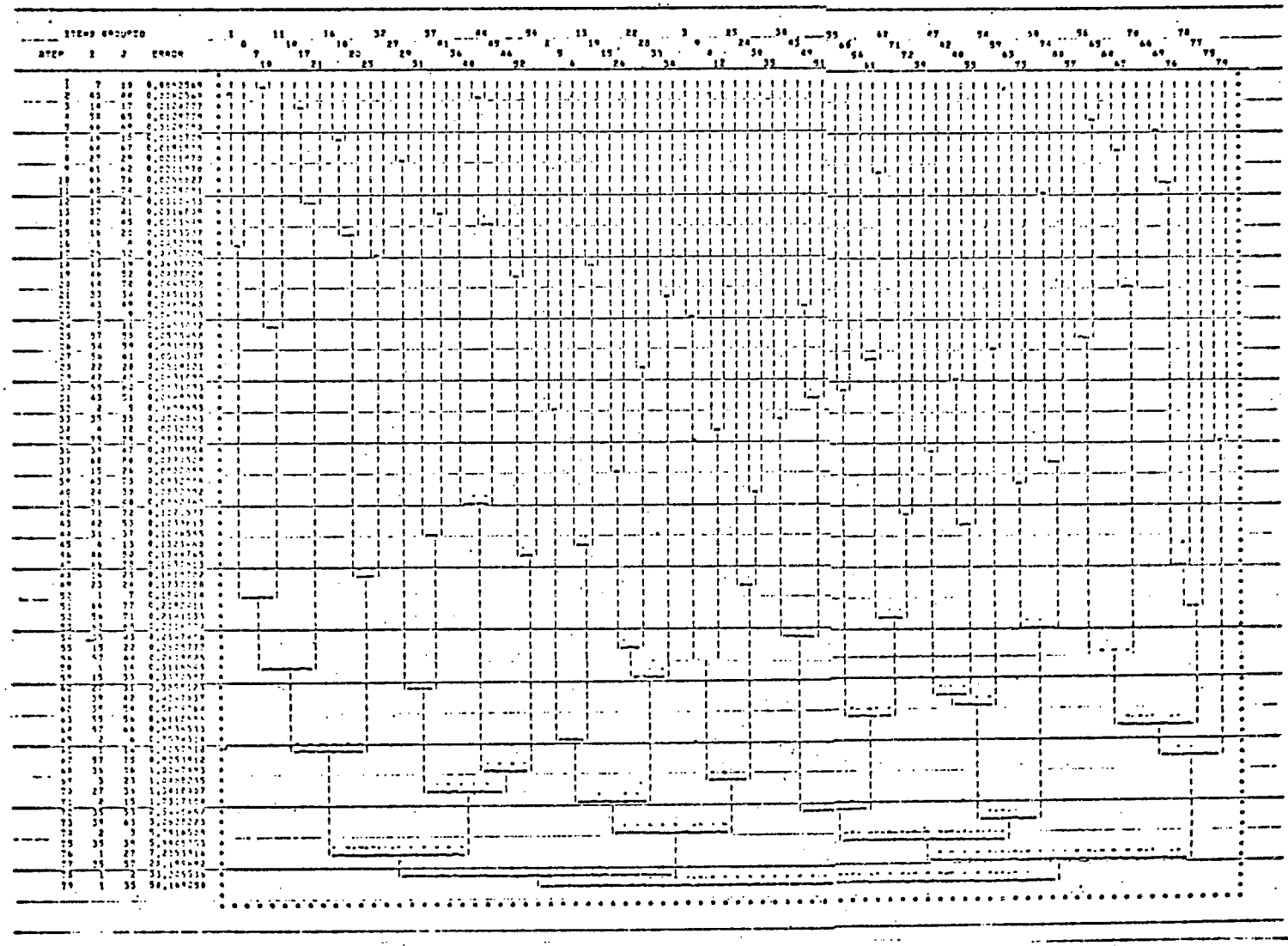
Sample Outputs from Program UBC:CGROUP

```
PROBLEM NAME **** CLUSTER TRIAL RUN 7 DATA1
NUMBER OF ITEMS TO BE GROUPED =  80
NUMBER OF GROUPING KEYS       =   2
START PRINTING WHEN THERE ARE    10 GROUPS
STANDARDIZE GROUPING KEYS : YES
PRINT A TREE GRAPH        : YES
CONTIGUITY CONSTRAINT     : NO
ITEM IDENTIFICATION NAMES TO BE READ : NO
STORE GROUP MEMBERSHIP    : NO
TRANSPOSE DATA MATRIX     : YES
NUMBER OF FORMAT CARDS    =  1
PLOT ERROR TERMS          : YES
   720 BYTES OF CORE ARE ACQUIRED TO TRANSPOSE THE DATA MATRIX
DATA FORMAT : (5X,2F7,0)
EXECUTION TIME FOR TRANSPOSING =  0,09 SECONDS
  17444 BYTES OF CORE ARE ACQUIRED FOR GROUPING
TIME TO READ DATA AND STORE ERROR MATRIX =    0,06 SECONDS
```

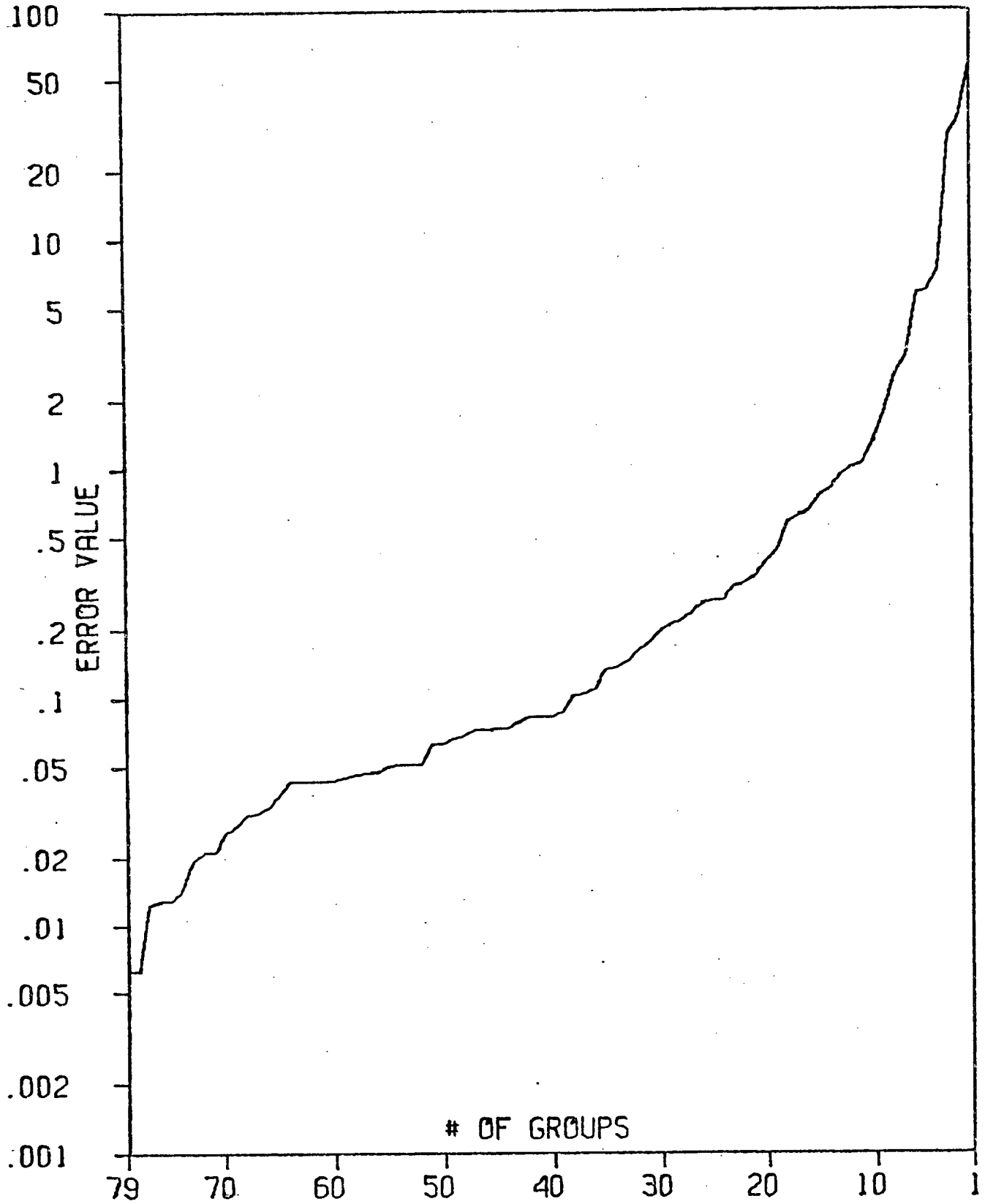
```
***WARNING*** OUTPUT FIELD WIDTH TOO SMALL. CONDITION OCCURRED DURING A FORMATTED WRITE ON FORTRAN UNIT 6 WHICH IS ATTACHED TO
               *SINK*. THE WRITE IS SEQUENTIAL AT RECORD NUMBER 21. FOR THIS AND ALL FUTURE OCCURRENCES OF THIS CONDITION, A
               FIELD OF *'S WILL BE WRITTEN.
```

```
STEP  1   79 GROUPS AFTER JOINING   7 (N=  1) &  10 (N=  1)  ERROR = 0,625688E-02CUM = 0,625688E-02INDEX = **********
STEP  2   78 GROUPS AFTER JOINING  40 (N=  1) &  44 (N=  1)  ERROR = 0,625693E-02CUM = 0,125138E-01INDEX =     0,0006
STEP  3   77 GROUPS AFTER JOINING  14 (N=  1) &  17 (N=  1)  ERROR = 0,124777E-01CUM = 0,249915E-01INDEX =    77,5492
STEP  4   76 GROUPS AFTER JOINING  58 (N=  1) &  65 (N=  1)  ERROR = 0,129779E-01CUM = 0,379694E-01INDEX =     3,0870
STEP  5   75 GROUPS AFTER JOINING  66 (N=  1) &  69 (N=  1)  ERROR = 0,129780E-01CUM = 0,509474E-01INDEX =     0,0001
STEP  6   74 GROUPS AFTER JOINING  16 (N=  1) &  18 (N=  1)  ERROR = 0,140779E-01CUM = 0,650253E-01INDEX =     6,3565
STEP  7   73 GROUPS AFTER JOINING  64 (N=  1) &  67 (N=  1)  ERROR = 0,191988E-01CUM = 0,842240E-01INDEX =    26,9177
STEP  8   72 GROUPS AFTER JOINING  27 (N=  1) &  29 (N=  1)  ERROR = 0,211870E-01CUM = 0,105411    INDEX =     7,5598
STEP  9   71 GROUPS AFTER JOINING  61 (N=  1) &  62 (N=  1)  ERROR = 0,211870E-01CUM = 0,126598    INDEX =     0,0
STEP 10   70 GROUPS AFTER JOINING  66 (N=  2) &  76 (N=  1)  ERROR = 0,256627E-01CUM = 0,152261    INDEX =    14,9986
STEP 11   69 GROUPS AFTER JOINING  68 (N=  1) &  74 (N=  1)  ERROR = 0,279081E-01CUM = 0,180169    INDEX =     6,1247
STEP 12   68 GROUPS AFTER JOINING  14 (N=  2) &  21 (N=  1)  ERROR = 0,310433E-01CUM = 0,211212    INDEX =     7,7516
STEP 13   67 GROUPS AFTER JOINING  37 (N=  1) &  41 (N=  1)  ERROR = 0,316809E-01CUM = 0,242893    INDEX =     1,3967
STEP 14   66 GROUPS AFTER JOINING  40 (N=  2) &  45 (N=  1)  ERROR = 0,333699E-01CUM = 0,276263    INDEX =     3,5719
STEP 15   65 GROUPS AFTER JOINING  16 (N=  2) &  20 (N=  1)  ERROR = 0,383507E-01CUM = 0,314613    INDEX =     9,5511
STEP 16   64 GROUPS AFTER JOINING   1 (N=  1) &   8 (N=  1)  ERROR = 0,432025E-01CUM = 0,357816    INDEX =     8,2232
STEP 17   63 GROUPS AFTER JOINING  25 (N=  1) &  32 (N=  1)  ERROR = 0,432025E-01CUM = 0,401018    INDEX =     0,0
STEP 18   62 GROUPS AFTER JOINING  13 (N=  1) &  19 (N=  1)  ERROR = 0,437028E-01CUM = 0,444721    INDEX =     0,7296
STEP 19   61 GROUPS AFTER JOINING  46 (N=  1) &  52 (N=  1)  ERROR = 0,437029E-01CUM = 0,488424    INDEX =     0,0001
STEP 20   60 GROUPS AFTER JOINING  64 (N=  2) &  70 (N=  1)  ERROR = 0,443252E-01CUM = 0,532749    INDEX =     0,8685
STEP 21   59 GROUPS AFTER JOINING  33 (N=  1) &  34 (N=  1)  ERROR = 0,451103E-01CUM = 0,577860    INDEX =     1,0628
STEP 22   58 GROUPS AFTER JOINING  43 (N=  1) &  49 (N=  1)  ERROR = 0,467868E-01CUM = 0,624646    INDEX =     2,1927
STEP 23   57 GROUPS AFTER JOINING   3 (N=  1) &   9 (N=  1)  ERROR = 0,473313E-01CUM = 0,671978    INDEX =     0,6749
STEP 24   56 GROUPS AFTER JOINING   7 (N=  2) &  11 (N=  1)  ERROR = 0,483002E-01CUM = 0,720278    INDEX =     1,1669
STEP 25   55 GROUPS AFTER JOINING  57 (N=  1) &  58 (N=  2)  ERROR = 0,505458E-01CUM = 0,770824    INDEX =     2,6035
STEP 26   54 GROUPS AFTER JOINING  54 (N=  1) &  59 (N=  1)  ERROR = 0,515873E-01CUM = 0,822411    INDEX =     1,1332
STEP 27   53 GROUPS AFTER JOINING  56 (N=  1) &  61 (N=  2)  ERROR = 0,516347E-01CUM = 0,874046    INDEX =     0,0496
STEP 28   52 GROUPS AFTER JOINING  22 (N=  1) &  28 (N=  1)  ERROR = 0,519121E-01CUM = 0,925958    INDEX =     0,2847
STEP 29   51 GROUPS AFTER JOINING  42 (N=  1) &  48 (N=  1)  ERROR = 0,631090E-01CUM = 0,989067    INDEX =    11,2159
STEP 30   50 GROUPS AFTER JOINING  55 (N=  1) &  60 (N=  1)  ERROR = 0,631093E-01CUM = 1,05218     INDEX =     0,0001
STEP 31   49 GROUPS AFTER JOINING  43 (N=  2) &  51 (N=  1)  ERROR = 0,668039E-01CUM = 1,11898     INDEX =     2,9271
STEP 32   48 GROUPS AFTER JOINING   2 (N=  1) &   5 (N=  1)  ERROR = 0,689653E-01CUM = 1,18794     INDEX =     1,5054
STEP 33   47 GROUPS AFTER JOINING  35 (N=  1) &  38 (N=  1)  ERROR = 0,724863E-01CUM = 1,26043     INDEX =     2,4506
STEP 34   46 GROUPS AFTER JOINING   4 (N=  1) &  12 (N=  1)  ERROR = 0,732555E-01CUM = 1,33368     INDEX =     0,4987
STEP 35   45 GROUPS AFTER JOINING  75 (N=  1) &  79 (N=  1)  ERROR = 0,739952E-01CUM = 1,40768     INDEX =     0,4465
STEP 36   44 GROUPS AFTER JOINING  39 (N=  1) &  47 (N=  1)  ERROR = 0,739954E-01CUM = 1,48167     INDEX =     0,5251
STEP 37   43 GROUPS AFTER JOINING  68 (N=  2) &  80 (N=  1)  ERROR = 0,794309E-01CUM = 1,56110     INDEX =     3,2321
STEP 38   42 GROUPS AFTER JOINING  15 (N=  1) &  26 (N=  1)  ERROR = 0,832088E-01CUM = 1,64431     INDEX =     2,0451
STEP 39   41 GROUPS AFTER JOINING  63 (N=  1) &  73 (N=  1)  ERROR = 0,832088E-01CUM = 1,72752     INDEX =     0,0
STEP 40   40 GROUPS AFTER JOINING  24 (N=  1) &  30 (N=  1)  ERROR = 0,832092E-01CUM = 1,81073     INDEX =     0,0002
STEP 41   39 GROUPS AFTER JOINING  36 (N=  1) &  40 (N=  3)  ERROR = 0,862960E-01CUM = 1,89702     INDEX =     1,4838
STEP 42   38 GROUPS AFTER JOINING  71 (N=  1) &  72 (N=  1)  ERROR = 0,102058  CUM = 1,99908  INDEX =     7,1232
STEP 43   37 GROUPS AFTER JOINING  42 (N=  2) &  53 (N=  1)  ERROR = 0,103961  CUM = 2,10304  INDEX =     0,7088
STEP 44   36 GROUPS AFTER JOINING  31 (N=  1) &  37 (N=  2)  ERROR = 0,109658  CUM = 2,21270  INDEX =     2,0276
STEP 45   35 GROUPS AFTER JOINING   6 (N=  1) &  13 (N=  2)  ERROR = 0,132346  CUM = 2,34505  INDEX =     7,4481
STEP 46   34 GROUPS AFTER JOINING  46 (N=  2) &  50 (N=  1)  ERROR = 0,134476  CUM = 2,47952  INDEX =     0,5634
STEP 47   33 GROUPS AFTER JOINING  66 (N=  3) &  78 (N=  1)  ERROR = 0,143061  CUM = 2,62258  INDEX =     2,1705
STEP 48   32 GROUPS AFTER JOINING  16 (N=  3) &  25 (N=  2)  ERROR = 0,161870  CUM = 2,78445  INDEX =     4,3386
```

```
STEP  49   31 GROUPS AFTER JOINING  23 (N=  13) &  24 (N=   2)  ERROR = 0,173719    CUM =  2,95517    INDEX =    2,3333
STEP  50   30 GROUPS AFTER JOINING   1 (N=  23) &   7 (N=   3)  ERROR = 0,196671    CUM =  3,15184    INDEX =    3,7500
STEP  51   29 GROUPS AFTER JOINING  66 (N=  43) &  77 (N=   1)  ERROR = 0,209681    CUM =  3,36208    INDEX =    2,2653
STEP  52   28 GROUPS AFTER JOINING  56 (N=  33) &  71 (N=  23)  ERROR = 0,214113    CUM =  3,57620    INDEX =    0,6753
STEP  53   27 GROUPS AFTER JOINING  63 (N=  23) &  68 (N=  33)  ERROR = 0,231662    CUM =  3,80780    INDEX =    2,2891
STEP  54   26 GROUPS AFTER JOINING  35 (N=  23) &  43 (N=  33)  ERROR = 0,254769    CUM =  4,06255    INDEX =    2,9984
STEP  55   25 GROUPS AFTER JOINING  15 (N=  23) &  22 (N=  23)  ERROR = 0,260579    CUM =  4,32312    INDEX =    0,5969
STEP  56   24 GROUPS AFTER JOINING  57 (N=  33) &  64 (N=  33)  ERROR = 0,261902    CUM =  4,58502    INDEX =    0,1771
STEP  57   23 GROUPS AFTER JOINING   3 (N=  23) &   4 (N=  23)  ERROR = 0,302529    CUM =  4,88755    INDEX =    3,7229
STEP  58   22 GROUPS AFTER JOINING   1 (N=  53) &  14 (N=  33)  ERROR = 0,311654    CUM =  5,19921    INDEX =    0,6938
STEP  59   21 GROUPS AFTER JOINING  15 (N=  43) &  33 (N=  23)  ERROR = 0,337306    CUM =  5,53227    INDEX =    1,5810
STEP  60   20 GROUPS AFTER JOINING  27 (N=  23) &  31 (N=  33)  ERROR = 0,389512    CUM =  5,92178    INDEX =    3,2535
STEP  61   19 GROUPS AFTER JOINING  39 (N=  23) &  42 (N=  33)  ERROR = 0,443764    CUM =  6,36554    INDEX =    2,3235
STEP  62   18 GROUPS AFTER JOINING  39 (N=  53) &  54 (N=  23)  ERROR = 0,566906    CUM =  6,92385    INDEX =    5,7749
STEP  63   17 GROUPS AFTER JOINING  55 (N=  23) &  56 (N=  53)  ERROR = 0,611869    CUM =  7,53531    INDEX =    1,4276
STEP  64   16 GROUPS AFTER JOINING  57 (N=  63) &  66 (N=  53)  ERROR = 0,653653    CUM =  8,18897    INDEX =    1,3609
STEP  65   15 GROUPS AFTER JOINING   2 (N=  23) &   6 (N=  33)  ERROR = 0,755431    CUM =  8,94328    INDEX =    2,4623
STEP  66   14 GROUPS AFTER JOINING   1 (N=  63) &  16 (N=  53)  ERROR = 0,805762    CUM =  9,74916    INDEX =    1,3222
STEP  67   13 GROUPS AFTER JOINING  57 (N=  11) &  75 (N=  23)  ERROR = 0,925361    CUM =  10,6736    INDEX =    2,0781
STEP  68   12 GROUPS AFTER JOINING  36 (N=  43) &  46 (N=  33)  ERROR = 1,00475     CUM =  11,6783    INDEX =    1,1169
STEP  69   11 GROUPS AFTER JOINING   3 (N=  43) &  23 (N=  53)  ERROR = 1,04682     CUM =  12,7251    INDEX =    0,5025

GROUP No  STEP  70   10 GROUPS AFTER JOINING  27 (N=   5) &  36 (N=   7)  ERROR = 1,36184     CUM =  14,0870    INDEX =    5,1000
  1 13      1  7  8 10 11 14 16 17 18 20 21 25 32
  2  5      2  5  6 13 19
  3  7      3  4  9 12 23 24 30
 15  6     15 22 26 28 33 34
 27 12     27 29 31 36 37 40 41 44 45 46 50 52
 35  5     35 38 43 49 51
 39  7     39 42 47 48 53 54 59
 55  7     55 56 60 61 62 71 72
 57 13     57 58 64 65 66 67 69 70 75 76 77 78 79
 63  5     63 68 73 74 80

GROUP No  STEP  71    9 GROUPS AFTER JOINING   2 (N=   5) &  15 (N=   6)  ERROR = 1,73171     CUM =  15,7987    INDEX =    2,9055
  1 13      1  7  8 10 11 14 16 17 18 20 21 25 32
  2 11      2  5  6 13 15 19 22 26 28 33 34
  3  7      3  4  9 12 23 24 30
 27 12     27 29 31 36 37 40 41 44 45 46 50 52
 35  5     35 38 43 49 51
 39  7     39 42 47 48 53 54 59
 55  7     55 56 60 61 62 71 72
 57 13     57 58 64 65 66 67 69 70 75 76 77 78 79
 63  5     63 68 73 74 80

GROUP No  STEP  72    8 GROUPS AFTER JOINING  35 (N=   5) &  55 (N=   7)  ERROR = 2,54457     CUM =  18,3432    INDEX =    4,2285
  1 13      1  7  8 10 11 14 16 17 18 20 21 25 32
  2 11      2  5  6 13 15 19 22 26 28 33 34
  3  7      3  4  9 12 23 24 30
 27 12     27 29 31 36 37 40 41 44 45 46 50 52
 35 12     35 38 43 49 51 55 56 60 61 62 71 72
 39  7     39 42 47 48 53 54 59
 57 13     57 58 64 65 66 67 69 70 75 76 77 78 79
 63  5     63 68 73 74 80

GROUP No  STEP  73    7 GROUPS AFTER JOINING  39 (N=   7) &  63 (N=   5)  ERROR = 3,06282     CUM =  21,4061    INDEX =    1,6294
  1 13      1  7  8 10 11 14 16 17 18 20 21 25 32
  2 11      2  5  6 13 15 19 22 26 28 33 34
  3  7      3  4  9 12 23 24 30
 27 12     27 29 31 36 37 40 41 44 45 46 50 52
 35 12     35 38 43 49 51 55 56 60 61 62 71 72
 39 12     39 42 47 48 53 54 59 63 68 73 74 80
 57 13     57 58 64 65 66 67 69 70 75 76 77 78 79

GROUP No  STEP  74    6 GROUPS AFTER JOINING   2 (N=  11) &   3 (N=   7)  ERROR = 5,79165     CUM =  27,1977    INDEX =    6,2367
  1 13      1  7  8 10 11 14 16 17 18 20 21 25 32
  2 18      2  3  4  5  6  9 12 13 15 19 22 23 24 26 28 30 33 34
 27 12     27 29 31 36 37 40 41 44 45 46 50 52
 35 12     35 38 43 49 51 55 56 60 61 62 71 72
 39 12     39 42 47 48 53 54 59 63 68 73 74 80
 57 13     57 58 64 65 66 67 69 70 75 76 77 78 79

GROUP No  STEP  75    5 GROUPS AFTER JOINING  35 (N=  12) &  39 (N=  12)  ERROR = 5,99059     CUM =  33,1883    INDEX =    0,2061
  1 13      1  7  8 10 11 14 16 17 18 20 21 25 32
  2 18      2  3  4  5  6  9 12 13 15 19 22 23 24 26 28 30 33 34
 27 12     27 29 31 36 37 40 41 44 45 46 50 52
 35 24     35 38 39 42 43 47 48 49 51 53 54 55 56 59 60 61 62 63 68 71 72 73 74 80
 57 13     57 58 64 65 66 67 69 70 75 76 77 78 79

GROUP No  STEP  76    4 GROUPS AFTER JOINING   1 (N=  13) &  27 (N=  12)  ERROR = 7,23539     CUM =  40,4237    INDEX =    1,0390
  1 25      1  7  8 10 11 14 16 17 18 20 21 25 27 29 31 32 36 37 40 41 44 45 46 50 52
  2 18      2  3  4  5  6  9 12 13 15 19 22 23 24 26 28 30 33 34
 35 24     35 38 39 42 43 47 48 49 51 53 54 55 56 59 60 61 62 63 68 71 72 73 74 80
 57 13     57 58 64 65 66 67 69 70 75 76 77 78 79

GROUP No  STEP  77    3 GROUPS AFTER JOINING  35 (N=  24) &  57 (N=  13)  ERROR = 28,1997     CUM =  68,6234    INDEX =    11,5899
  1 25      1  7  8 10 11 14 16 17 18 20 21 25 27 29 31 32 36 37 40 41 44 45 46 50 52
  2 18      2  3  4  5  6  9 12 13 15 19 22 23 24 26 28 30 33 34
 35 37     35 38 39 42 43 47 48 49 51 53 54 55 56 57 58 59 60 61 62 63 68 69 70 71 72 73
           74 75 76 77 78 79 80

GROUP No  STEP  78    2 GROUPS AFTER JOINING   1 (N=  25) &   2 (N=  18)  ERROR = 33,2055     CUM =  101,829    INDEX =    0,5325
  1 43      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
           31 32 33 34 36 37 40 41 44 45 46 50 52
 35 37     35 38 39 42 43 47 48 49 51 53 54 55 56 57 58 59 60 61 62 63 68 69 70 71 72 73
           74 75 76 77 78 79 80

GROUP No  STEP  79    1 GROUPS AFTER JOINING   1 (N=  43) &  35 (N=  37)  ERROR = 58,1693     CUM =  159,998    INDEX =    1,5036
  1 80      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
           31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
           61 62 63 68 69 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

EXECUTION TIME FOR GROUPING =    0,26 SECONDS
```



LUSTER TRIAL RUN / DATA1



APPENDIX E

Listing and Sample Outputs from NONHIER: a Computer Program for Three Nonhierarchical Clustering Techniques

```
      DIMENSION X(7500)
      LIMIT=7500
      CALL EXEC(X,LIMIT)
      STOP
      END
      SUBROUTINE EXEC(X,LIMIT)
C
C   THIS SUBROUTINE READS PARAMETERS, COMPUTES STORAGE AND CALLS MAJOR
C   PROGRAM SEGMENTS NEEDED FOR A NON-HIERARCHICAL CLUSTERING JOB USING
C   ONE OF THE METHODS PROGRAMMED AS A VERSION OF SUBROUTINE *KMEAN*.
C
C   EVERY JOB REQUIRES THREE USER SUPPLIED DECK SEGMENTS.
C
C   1. PROGRAM *DRIVER* PERFORMS THE FOLLOWING TASKS.
C      A. ASSIGNS INPUT/OUTPUT UNITS.
C      B. ESTABLISHES THE DIMENSION OF THE *X* ARRAY AND SETS THIS
C         DIMENSION TO *LIMIT*
C      C. CALLS SUBROUTINE *EXEC*.
C   THE FOLLOWING EXAMPLE WILL SUFFICE IN MOST CASES.
C
C      PROGRAM DRIVER(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
C     ATAPE7=PUNCH,TAPE1,TAPE2)
C      DIMENSION X(5000)
C      LIMIT=5000
C      CALL EXEC(X,LIMIT)
C      END
C
C   2. SUBROUTINE *USER* IS EMPLOYED TO READ THE COMPLETE SET OF SCORES
C      ON THE VARIABLES FOR ONE DATA UNIT. THE FOLLOWING EXAMPLE
C      ILLUSTRATES VARIOUS POSSIBILITIES FOR MERGING FILES AND
C      TRANSFORMING VARIABLES AS THEY ARE READ.
C
C      SUBROUTINE USER(X)
C      DIMENSION X(8)
C      READ(1,100) X(7),Y
C      READ(2) (X(I),I=1,6)
C      READ(5,200) X(8),Z
C      X(3)=.5*X(3)
C      X(7)=3.6*X(7)
C      X(8)=.4*X(8)+.35*Y+.25*Z*X(8)
C      RETURN
C  100 FORMAT(2F11.3)
C  200 FORMAT(F8.1,F6.3)
C      END
C
C   3. FUNCTION *DIST* COMPUTES THE DISTANCE BETWEEN TWO DATA UNITS OR
C      BETWEEN A DATA UNIT AND A CLUSTER CENTROID. THE USER CAN SPECIFY
C      ANY DESIRED DISTANCE FUNCTION AND WEIGHT THE VARIABLES IN ANY
C      MANNER.  THE FOLLOWING EXAMPLE ILLUSTRATES A WEIGHTED SQUARED
```

```
C      EUCLIDEAN DISTANCE BETWEEN TWO DATA UNITS DENOTED AS X AND Y.
C      THE PROBLEM INVOLVES 8 VARIABLES AND THE WEIGHTS ARE IN THE
C      *W* ARRAY.
C
C      FUNCTION DIST(X,Y)
C      DIMENSION X(1),Y(1),W(8)
C      DATA (W(I),I=1,8)/3*1.,3.,4.5,2.,2*1/
C      DIST=0.
C      DO 10 I=1,8
C   10 DIST=DIST+W(I)*((X(I)-Y(I))**2)
C      RETURN
C      END
C
C   NOTE THAT SCALING AND TRANSFORMATION OF VARIABLES CAN BE
C   ACCOMPLISHED EITHER IN SUBROUTINE *USER* OR IN SUBROUTINE *DIST*.
C
C-----------------------------------------------------------------------
C   INPUT SPECIFICATIONS
C   CARD 1  TITLE
C   CARD 2  PARAMETER CARD
C     COLS  1- 5  NE=NUMBER OF ENTITIES (DATA UNITS)
C     COLS  6-10  NV=NUMBER OF VARIABLES
C     COLS 11-15  NC=NUMBER OF CLUSTERS
C     COLS 16-20  NTIN=INPUT UNIT FOR THE DATA SET
C                 NTIN=5, CARD READER
C                 NTIN.NE.5, TAPE OR DISK FILE
C     COLS 21-25  NTOUT=OUTPUT UNIT FOR SAVING CLUSTER MEMBERSHIP LISTS
C                 NTOUT=7, CARD PUNCH
C                 NTOUT.LE.0, DO NOT SAVE MEMBERSHIP LISTS
C     COLS 26-30  MINREL=TERMINATION PARAMETER.  CLUSTERING ENDS WHEN A
C                        CYCLE THROUGH THE DATA SET RESULTS IN *MINREL*
C                        OR FEWER CHANGES IN CLUSTER MEMBERSHIPS
C                 MINREL.LE.0, ITERATE TO COMPLETE CONVERGENCE
C     COLS 31-35  IPART=INITIAL PARTITION PARAMETER
C                 IPART=1, SEED POINTS ARE SELECTED FROM THE DATA UNITS.
C                          READ THE SEQUENCE NUMBERS FOR THE CHOSEN DATA
C                          UNITS FROM CARD(S) 3 IN 2014 FORMAT.  IF THE
C                          DATA SET IS NOT STORED IN CORE, THE LIST OF
C                          OF SEQUENCE NUMBERS MUST BE IN ASCENDING ORDER
C                 IPART=2, THE DATA UNITS ARE GROUPED INTO AN INITIAL
C                          PARTITION IN THE INPUT SEQUENCE WITH THE
C                          FIRST *NUMBR(1)* IN CLUSTER 1, THE NEXT
C                          *NUMBR(2)* IN CLUSTER 2 ETC.  READ THE
C                          *NUMBR* ARRAY FROM CARD(S) 3 IN 2014 FORMAT.
C                 IPART=3, THE SCORE VECTORS FOR THE SEED POINTS ARE
C                          READ FROM CARD(S) 4 IN FORMAT *FMT* WHICH IS
C                          READ FROM CARD 3.
C     COLS 36-40 METHOD=PARAMETER FOR CHOOSING THE ALGORITHM IN ONE
C                       VERSION OF SUBROUTINE *KMEAN*.
C                 METHOD=1, JANCEY ALGORITHM
```

```
C                 METHOD.NE.1, FORGY ALGORITHM
C
C***CARDS 3 AND 4 ARE READ IN SUBROUTINE *KMEAN* ACCORDING TO THE
C***PROCEDURE SPECIFIED BY THE CHOSEN VALUE OF *IPART*.  NOTE THAT THE
C***BASIC K-MEANS METHOD OF MACQUEEN SIMPLY USES THE FIRST *NC* DATA
C***UNITS AS CLUSTER SEED POINTS AND THEREFORE IGNORES THE *IPART*
C***PARAMETER.
C-----------------------------------------------------------------------
C
C  STORAGE ALLOCATIONS IN THE *X* ARRAY
C  X(N1) TO X(N2-1)  NC*NV WORDS--STORAGE OF THE CENTR ARRAY
C  X(N2) TO X(N3-1)  NC WORDS--STORAGE OF THE NUMBR ARRAY
C  X(N3) TO X(N4-1)  NE WORDS--STORAGE OF THE MEMBR ARRAY
C  X(N4) TO X(N5-1)  NC*NV WORDS--STORAGE OF THE TOTAL ARRAY
C  X(N5) TO X(N6)    NV OR NE*NV WORDS--STORAGE OF THE DATA ARRAY
C  X(N4) TO X(N7)    NE WORDS--STORAGE OF THE LIST ARRAY IN *RESULT*
C
      DIMENSION X(1),TITLE(20)
      READ(5,1000) TITLE
      READ(5,1100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
      WRITE(6,2000) TITLE
      WRITE(6,2100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
      N1=1
      N2=N1+NC*NV
      N3=N2+NC
      N4=N3+NE
      N5=N4+NC*NV
C  *N6* MAY BE INCREASED IN *KMEAN*.
      N6=N5+NV-1
      N7=N4+NE-1
      MAX=N6
      IF(N7.GT.MAX) MAX=N7
      WRITE(6,2200) MAX,LIMIT
      IF(MAX.GT.LIMIT) STOP
      CALL KMEAN(X(N1),X(N2),X(N3),X(N4),X(N5),N5,NE,NV,NC,NTIN,MINREL,
     AIPART,METHOD,LIMIT)
      CALL RESULT(X(N1),X(N2),X(N3),X(N4),TITLE,NE,NV,NC,NTOUT)
      RETURN
1000 FORMAT(20A4)
1100 FORMAT(8I5)
2000 FORMAT(1H1,20A4)
2100 FORMAT(5HONE =,I8,/,5H NV =,I8,/,5H NC =,I8,/,7H NTIN =,I6,/,
     A8H NTOUT =,I5,/,9H MINREL =,I4,/,8H IPART =,I5,/,9H METHOD =,I4)
2200 FORMAT(19HOREQUIRED STORAGE =,I5,6H WORDS,/,
     A       19HOALLOTTED STORAGE =,I5,6H WORDS)
      END
C
      SUBROUTINE RESULT(CENTR,NUMBR,MEMBR,LIST,TITLE,NE,NV,NC,NTOUT)
C  THIS SUBROUTINE PRINTS THE RESULTS FROM A CLUSTERING JOB BASED
C  ON ANY VERSION OF SUBROUTINE *KMEAN*.
```

```
C
      DIMENSION CENTR(1),NUMBR(1),MEMBR(1),LIST(1),TITLE(20)
C
C   AS A CONTINGENCY PRECAUTION WRITE OUT THE RAW MEMBERSHIP LIST.
      WRITE(6,2000) TITLE
      WRITE(6,2100) (MEMBR(K),K=1,NE)
      WRITE(6,2200) (NUMBR(J),J=1,NC)
C   INVERT THE *MEMBR* ARRAY AND PUT THE RESULT IN THE *LIST* ARRAY.
C   FIRST REVISE THE *NUMBR* ARRAY TO CONTAIN START POINTS IN THE
C   *LIST* ARRAY FOR EACH CLUSTER
      NUMBR(NC)=NE-NUMBR(NC)+1
      JJ=NC
      JJ1=JJ-1
      DO 10 J=2,NC
      NUMBR(JJ1)=NUMBR(JJ)-NUMBR(JJ1)
      JJ=JJ1
10    JJ1=JJ-1
C   BUILD *LIST* ARRAY
      DO 20 K=1,NE
      MEMBRK=MEMBR(K)
      NJ=NUMBR(MEMBRK)
      LIST(NJ)=K
      NUMBR(MEMBRK)=NUMBR(MEMBRK)+1
20    CONTINUE
C   SAVE THE SORTED MEMBERSHIP LIST IF DESIRED
      IF(NTOUT.LE.0) GO TO 30
      WRITE(NTOUT,3000) TITLE
      WRITE(NTOUT,3100) (LIST(K),K=1,NE)
C   RESTORE THE *NUMBR* ARRAY
30    JJ=NC
      DO 40 J=2,NC
      NUMBR(JJ)=NUMBR(JJ)-NUMBR(JJ-1)
40    JJ=JJ-1
      NUMBR(1)=NUMBR(1)-1
C   PRINT RESULTS FOR EACH CLUSTER
      WRITE(6,2000) TITLE
      WRITE(8,2000) TITLE
      K1=1
      DO 50 J=1,NC
      WRITE(6,2300) J,NUMBR(J)
      WRITE(8,2301) J,NUMBR(J)
      J1=(J-1)*NV
      WRITE(6,2400) (CENTR(J1+I),I=1,NV)
      K2=K1+NUMBR(J)-1
      WRITE(6,2500) (LIST(K),K=K1,K2)
      WRITE(8,2501) (LIST(KF),KF=K1,K2)
      K1=K2+1
50    CONTINUE
      WRITE(6,3500)
      RETURN
```

```
2000 FORMAT(1H1,20A4)
2100 FORMAT(20HORAW MEMBERSHIP LIST,/,(1X,25I5))
2200 FORMAT(14HOCLUSTER SIZES,/,(1X,25I5))
2300 FORMAT(8HOCLUSTER,I3,9H CONTAINS,I5,11H DATA UNITS)
2301 FORMAT(2I4)
2400 FORMAT(21HOCENTROID COORDINATES,/,(1X,10E12.4))
2500 FORMAT(16HOMEMBERSHIP LIST,/,(1X,25I5))
2501 FORMAT(15I5)
3000 FORMAT(20A4)
3100 FORMAT(20I4)
3500 FORMAT('1',15X,'END OF OUTPUT',///)
     END
C
     SUBROUTINE USER(X)
     DIMENSION X(2)
     READ(5,100) X(1),X(2)
     RETURN
100  FORMAT(5X,2F10.2)
     END
C
     FUNCTION DIST(X,Y)
     DIMENSION X(1),Y(1)
     DIST=0.
     DO 10 I=1,2
10   DIST=DIST+((X(I)-Y(I))**2)
     RETURN
     END
$SIG
```

```
      SUBROUTINE KMEAN(CENTR,NUMBR,MEMBR,TOTAL,DATA,N5,NE,NV,NC,NTIN,
     AMINREL,IPART,METHOD,LIMIT)
C
C-----------------------------------------------------------------------
C  VERSION 1.  THE DATA SET IS STORED IN CENTRAL MEMORY.
C-----------------------------------------------------------------------
C
C  THIS SUBROUTINE ITERATIVELY SORTS *NE* DATA UNITS INTO *NC* CLUSTERS
C  USING THE ALGORITHM OF (METHOD.NE.1)
C
C  FORGY, E.W., CLUSTER ANALYSIS OF MULTIVARIATE DATA, EFFICIENCY
C  VERSUS INTERPRETABILITY OF CLASSIFICATIONS, PAPER PRESENTED AT THE
C  BIOMETRIC SOCIETY (WNAR) MEETINGS, RIVERSIDE, CALIFORNIA, JUNE
C  1965.  ABSTRACT IN BIOMETRICS, VOLUME 21, NUMBER 3, P 768.
C
C  OR THE ALGORITHM OF (METHOD=1)
C
C  JANCEY, R.C., MULTIDIMENSIONAL GROUP ANALYSIS, AUSTRALIAN JOURNAL
C  OF BOTANY, VOLUME 14, NUMBER 1, APRIL 1966, PP 127-130.
C
C  CENTR(NV*(J-1)+I)=SCORE ON I-TH VARIABLE FOR J-TH CLUSTER CENTROID
C  TOTAL(NV*(J-1)+I)=TOTAL SCORE ON I-<H VARIABLE FOR DATA UNITS THUS
C                    FAR ALLOCATED TO THE J-TH CLUSTER
C  NUMBR(J)=NUMBER OF DATA UNITS THUS FAR ALLOCATED TO THE J-TH CLUSTER
C  MEMBR(K)=CLUSTER TO WHICH THE K-TH DATA UNIT CURRENTLY BELONGS
C  DATA(NV*(K-1)+I)=SCORE ON I-TH VARIABLE FOR K-TH DATA UNIT
C
      DIMENSION CENTR(1),TOTAL(1),NUMBR(1),MEMBR(1),DATA(1),FMT(20)
     A,NAME(4)
      DATA (NAME(I),I=1,4)/4H    F,4HORGY,4H  JA,4HNCEY/
      I=1
      IF(METHOD.EQ.1) I=3
      WRITE(6,2000) NAME(I),NAME(I+1)
      WRITE(8,2001) NAME(I),NAME(I+1)
C  CHECK FOR SUFFICIENT STORAGE
      N6=N5+NE*NV-1
      WRITE(6,2100) N6,LIMIT
      IF(N6.GT.LIMIT) STOP
C  ESTABLISH INITIAL PARTITION
      IF(IPART.NE.3) GO TO 20
C  SEED POINTS ARE READ DIRECTLY FROM CARDS
      READ (5,1000) FMT
      WRITE(6,2200) FMT
      WRITE(6,2300)
      J1=0
      DO 10 J=1,NC
      READ(5,FMT) (CENTR(J1+I),I=1,NV)
      WRITE(6,2400) (CENTR(J1+I),I=1,NV)
10    J1=J1+NV
```

```
      GO TO 30
C   IPART=1 OR 2
20    WRITE(6,2500) IPART
      READ(5,1100) (NUMBR(J),J=1,NC)
      WRITE(6,2600) (NUMBR(J),J=1,NC)
C   READ THE DATA SET INTO CENTRAL MEMORY
30     K1=1
      DO 40 K=1,NE
      CALL USER (DATA(K1))
40    K1=K1+NV
      IF(IPART.EQ.3) GO TO 100
C   IF *IPART* IS 1 OR 2 SET UP THE SEED POINTS
      IF(IPART.EQ.2) GO TO 60
C   IPART=1.  THE DATA UNIT WITH SEQUENCE NUMBER *NUMBR(J)* IS USED AS
C   THE J-TH SEED POINT
      DO 50 J=1,NC
      NJ=(NUMBR(J)-1)*NV
      J1=(J-1)*NV
      DO 50 I=1,NV
      CENTR(J1+I)=DATA(NJ+I)
50    CONTINUE
      GO TO 100
C   IPART=2.  THE DATA UNITS ARE GROUPED INTO CLUSTERS WITH THE J-TH
C   CLUSTER HAVING *NUMBR(J)* MEMBERS.
60    K=0
      J1=-NV
C   ACCUMULATE THE TOTAL SCORE ON EACH VARIABLE FOR EACH CLUSTER
      DO 80 J=1,NC
      NJ=NUMBR(J)
      J1=J1+NV
      DO 70 I=1,NV
70    TOTAL(J1+I)=0.
      DO 80 KJ=1,NJ
      K=K+1
      MEMBR(K)=J
      K1=(K-1)*NV
      DO 80 I=1,NV
      J2=J1+I
      TOTAL(J2)=TOTAL(J2)+DATA(K1+I)
80    CONTINUE
C   COMPUTE THE CENTROIDS
      J1=0
      DO 90 J=1,NC
      DO 90 I=1,NV
      J1=J1+1
      CENTR(J1)=TOTAL(J1)/NUMBR(J)
90    CONTINUE
      GO TO 115
C   INITIALIZE ARRAYS
100   DO 110 K=1,NE
```

```
110   MEMBR(K)=0
115   NPASS=1
C  BEGINNING OF MAIN LOOP
120   J1=0
      DO 130 J=1,NC
      NUMBR(J)=0
      DO 130 I=1,NV
      J1=J1+1
130   TOTAL(J1)=0.
      MOVES=0
      TDIST=0
C  ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID
      K1=0
      DO 160 K=1,NE
      K2=K1+1
      J2=1
C  COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
      DREF=DIST(DATA(K2),CENTR(J2))
      JREF=1
C  TEST DISTANCES TO REMAINING CLUSTER CENTROIDS
      DO 140 J=2,NC
      J2=J2+NV
      DTEST=DIST(DATA(K2),CENTR(J2))
      IF(DTEST.GE.DREF) GO TO 140
      DREF=DTEST
      JREF=J
140   CONTINUE
C  ALLOCATE DATA UNIT *K* TO CLUSTER *JREF*
      NUMBR(JREF)=NUMBR(JREF)+1
      TDIST=TDIST+DREF
      IF(JREF.EQ.MEMBR(K)) GO TO 150
C  THE DATA UNIT CHANGES ITS MEMBERSHIP
      MOVES=MOVES+1
      MEMBR(K)=JREF
150   J1=(JREF-1)*NV
      DO 160 I=1,NV
      J1=J1+1
      K1=K1+1
      TOTAL(J1)=TOTAL(J1)+DATA(K1)
160   CONTINUE
C  ALL DATA UNITS ALLOCATED.  TEST FOR CONVERGENCE
      WRITE(6,2700) MOVES,NPASS,TDIST
      NPASS=NPASS+1
      JREF=0
      IF(MOVES.GT.MINREL) GO TO 185
      IF(METHOD.NE.1.AND.MOVES.EQ.0) RETURN
      JREF=1
C  COMPUTE TRUE CLUSTER CENTROIDS--FORGY UPDATE
170   J1=0
      DO 180 J=1,NC
```

```
      DO 180 I=1,NV
      J1=J1+1
180   CENTR(J1)=TOTAL(J1)/NUMBR(J)
      IF(JREF.EQ.1) RETURN
      GO TO 120
185   IF(METHOD.NE.1) GO TO 170
C  JANCEY UPDATE
190   J1=0
      DO 200 J=1,NC
      DO 200 I=1,NV
      J1=J1+1
200   CENTR(J1)=2.*TOTAL(J1)/NUMBR(J)-CENTR(J1)
      GO TO 120
1000  FORMAT(20A4)
1100  FORMAT(20I4)
2000  FORMAT(1H0,2A4, 53H METHOD OF CLUSTER ANALYSIS.  DATA SET STORED I
     AN CORE)
2001  FORMAT(20A4)
2100  FORMAT(19HOREQUIRED STORAGE =,I5,6H WORDS,/,
     A       19HOALLOTTED STORAGE =,I5,6H WORDS)
2200  FORMAT(7HOFORMAT,20A4)
2300  FORMAT( 43H1INITIAL CLUSTER CENTERS READ IN AS FOLLOWS///)
2400  FORMAT(1X,10E12.4)
2500  FORMAT(  9H1 IPART =,I2, 30H,  NUMBR ARRAY READ AS FOLLOWS///)
2600  FORMAT(1X,10I7)
2700  FORMAT(1H0,I5,37H DATA UNITS MOVED ON ITERATION NUMBER,I3,/,
     A38H SUMMED DEVIATIONS ABOUT SEED POINTS =,E16.8)
      END
```

```
      SUBROUTINE KMEAN(CENTR,NUMBR,MEMBR,TOTAL,DATA,N5,NE,NV,NC,NTIN,
     AMINREL,IPART,METHOD,LIMIT)
C
C-----------------------------------------------------------------------
C  VERSION 1.  THE DATA SET IS STORED IN CENTRAL MEMORY.
C-----------------------------------------------------------------------
C
C  THIS SUBROUTINE ITERATIVELY SORTS *NE* DATA UNITS INTO *NC* CLUSTERS
C  USING THE CONVERGENT K-MEANS METHOD DESCRIBED IN SECTION 7.2.2.
C
C  CENTR(NV*(J-1)+I)=SCORE ON I-TH VARIABLE FOR J-TH CLUSTER CENTROID
C  TOTAL(NV*(J-1)+I)=TOTAL SCORE ON I-TH VARIABLE FOR DATA UNITS THUS
C                    FAR ALLOCATED TO THE J-TH CLUSTER
C  NUMBR(J)=NUMBER OF DATA UNITS THUS FAR ALLOCATED TO THE J-TH CLUSTER
C  MEMBR(K)=CLUSTER TO WHICH THE K-TH DATA UNIT CURRENTLY BELONGS
C  DATA(NV*(K-1)+I)=SCORE ON I-TH VARIABLE FOR K-TH DATA UNIT
C
      DIMENSION CENTR(1),TOTAL(1),NUMBR(1),MEMBR(1),DATA(1),FMT(20)
      WRITE(6,2000)
      WRITE(8,2000)
C  CHECK FOR SUFFICIENT STORAGE
      N6=N5+NE*NV-1
      WRITE(6,2100) N6,LIMIT
      IF(N6.GT.LIMIT) STOP
C  ESTABLISH INITIAL PARTITION
      IF(IPART.NE.3) GO TO 20
C  SEED POINTS ARE READ DIRECTLY FROM CARDS
      READ (5,1000) FMT
      WRITE(6,2200) FMT
      WRITE(6,2300)
      J1=0
      DO 10 J=1,NC
      READ(5,FMT) (CENTR(J1+I),I=1,NV)
      WRITE(6,2400) (CENTR(J1+I),I=1,NV)
10    J1=J1+NV
      GO TO 30
C  IPART=1 OR 2
20    WRITE(6,2500) IPART
      READ(5,1100) (NUMBR(J),J=1,NC)
      WRITE(6,2600) (NUMBR(J),J=1,NC)
C  READ THE DATA SET INTO CENTRAL MEMORY
30    K1=1
      DO 40 K=1,NE
      CALL USER (DATA(K1))
40    K1=K1+NV
      IF(IPART.EQ.3) GO TO 51
C  IF *IPART* IS 1 OR 2 SET UP THE SEED POINTS
      IF(IPART.EQ.2) GO TO 60
C  IPART=1.  THE DATA UNIT WITH SEQUENCE NUMBER *NUMBR(J)* IS USED AS
C  THE J-TH SEED POINT
```

```
      DO 50 J=1,NC
      NJ=(NUMBR(J)-1)*NV
      J1=(J-1)*NV
      DO 50 I=1,NV
      CENTR(J1+I)=DATA(NJ+I)
50    CONTINUE
C   THE INITIAL CONFIGURATION IS GIVEN IN TERMS OF SEED POINTS.
C   CONSTRUCT AN INITIAL PARTITION BY ASSIGNING EACH DATA UNIT TO THE
C   NEAREST SEED POINT.  SEED POINTS REMAIN FIXED THROUGHOUT ASSIGNMENT
C   OF THE FULL DATA SET.
51    DO 52 K=1,NE
52    MEMBR(K)=0
      J1=0
      DO 53 J=1,NC
      NUMBR(J)=0
      DO 53 I=1,NV
      J1=J1+1
53    TOTAL(J1)=0.
C   ALLOCATE EACH DATA UNIT TO THE NEAREST SEED POINT
      K1=0
      DO 55 K=1,NE
      K2=K1+1
      J2=1
C   COMPUTE DISTANCE TO FIRST SEED POINT
      DREF=DIST(DATA(K2),CENTR(J2))
      JREF=1
C   TEST DISTANCES TO REMAINING SEED POINTS
      DO 54 J=2,NC
      J2=J2+NV
      DTEST=DIST(DATA(K2),CENTR(J2))
      IF(DTEST.GE.DREF) GO TO 54
      DREF=DTEST
      JREF=J
54    CONTINUE
C   ALLOCATE DATA UNIT *K* TO CLUSTER *JREF*
      NUMBR(JREF)=NUMBR(JREF)+1
      MEMBR(K)=JREF
      J1=(JREF-1)*NV
      DO 55 I=1,NV
      J1=J1+1
      K1=K1+1
      TOTAL(J1)=TOTAL(J1)+DATA(K1)
55    CONTINUE
      GO TO 85
C   IPART=2.  THE DATA UNITS ARE GROUPED INTO CLUSTERS WITH THE J-TH
C   CLUSTER HAVING *NUMBR(J)* MEMBERS.
60    K=0
      J1=-NV
C   ACCUMULATE THE TOTAL SCORE ON EACH VARIABLE FOR EACH CLUSTER
      DO 80 J=1,NC
```

```
      NJ=NUMBR(J)
      J1=J1+NV
      DO 70 I=1,NV
70    TOTAL(J1+I)=0.
      DO 80 KJ=1,NJ
      K=K+1
      MEMBR(K)=J
      K1=(K-1)*NV
      DO 80 I=1,NV
      J2=J1+I
      TOTAL(J2)=TOTAL(J2)+DATA(K1+I)
80    CONTINUE
C  COMPUTE THE CENTROIDS
85    J1=0
      DO 90 J=1,NC
      DO 90 I=1,NV
      J1=J1+1
      CENTR(J1)=TOTAL(J1)/NUMBR(J)
90    CONTINUE
C  INITIALIZE ARRAYS
100   NPASS=1
C  BEGINNING OF MAIN LOOP
120   MOVES=0
      TDIST=0
C  ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID
      K1=0
      DO 160 K=1,NE
      K2=K1+1
      J2=1
C  COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
      DREF=DIST(DATA(K2),CENTR(J2))
      JREF=1
C  TEST DISTANCES TO REMAINING CLUSTER CENTROIDS
      DO 140 J=2,NC
      J2=J2+NV
      DTEST=DIST(DATA(K2),CENTR(J2))
      IF(DTEST.GE.DREF) GO TO 140
      DREF=DTEST
      JREF=J
140   CONTINUE
      TDIST=TDIST+DREF
      IF(JREF.NE.MEMBR(K)) GO TO 155
      K1=K1+NV
      GO TO 160
C  REALLOCATE DATA UNIT*K* FROM CLUSTER *MEMBR(K)* TO CLUSTER *JREF*
155   MOVES=MOVES+1
      J2=MEMBR(K)
      NUMBR(J2)=NUMBR(J2)-1
      NUMBR(JREF)=NUMBR(JREF)+1
      MEMBR(K)=JREF
```

```
      J1=(JREF-1)*NV
      J3=(J2-1)*NV
      DO 150 I=1,NV
      J1=J1+1
      J3=J3+1
      K1=K1+1
      TOTAL(J1)=TOTAL(J1)+DATA(K1)
      CENTR(J1)=TOTAL(J1)/NUMBR(JREF)
      TOTAL(J3)=TOTAL(J3)-DATA(K1)
      CENTR(J3)=TOTAL(J3)/NUMBR(J2)
150   CONTINUE
160   CONTINUE
C   ALL DATA UNITS ALLOCATED.  TEST FOR CONVERGENCE
      WRITE(6,2700) MOVES,NPASS,TDIST
      NPASS=NPASS+1
      IF(MOVES.LE.MINREL) RETURN
      GO TO 120
1000  FORMAT(20A4)
1100  FORMAT(20I4)
2000  FORMAT( 46HOCONVERGENT K-MEANS METHOD OF CLUSTER ANALYSIS,/,
     A 24H DATA SET STORED IN CORE)
2100  FORMAT(19HOREQUIRED STORAGE =,I5,6H WORDS,/,
     A       19HOALLOTTED STORAGE =,I5,6H WORDS)
2200  FORMAT(7HOFORMAT,20A4)
2300  FORMAT( 43H1INITIAL CLUSTER CENTERS READ IN AS FOLLOWS///)
2400  FORMAT(1X,10E12.4)
2500  FORMAT(  9H1 IPART =,I2, 30H,  NUMBR ARRAY READ AS FOLLOWS///)
2600  FORMAT(1X,10I7)
2700  FORMAT(1HO,I5,37H DATA UNITS MOVED ON ITERATION NUMBER,I3,/,
     A38H SUMMED DEVIATIONS ABOUT SEED POINTS =,E16.8)
      END
$SIG
```

```
CLUSTER TRIAL RUN - NON-HIERARCHICAL - TDATA1.

 NE =       80
 NV =        2
 NC =        2
 NTIN =      5
 NTOUT =    -1
 MINREL =    1
 IPART =     2
 METHOD =    3

 REQUIRED STORAGE =  166 WORDS

 ALLOTTED STORAGE = 7500 WORDS

    FORGY METHOD OF CLUSTER ANALYSIS,  DATA SET STORED IN CORE

 REQUIRED STORAGE =  250 WORDS

 ALLOTTED STORAGE = 7500 WORDS


 IPART = 2,  NUMBR ARRAY READ AS FOLLOWS


     40    40

   37 DATA UNITS MOVED ON ITERATION NUMBER  1
 SUMMED DEVIATIONS ABOUT SEED POINTS =  0.29326290E 13

    1 DATA UNITS MOVED ON ITERATION NUMBER  2
 SUMMED DEVIATIONS ABOUT SEED POINTS =  0.68569030E 12


 RAW MEMBERSHIP LIST
    1   2   2   2   2   1   1   1   2   1   1   2   1   1   2   1   1   1   1   1   1   2   2   2   1
    2   1   2   1   2   1   1   2   2   2   1   1   2   1   1   1   2   2   1   1   1   1   2   2   1
    2   1   2   1   2   2   1   1   2   2   2   2   2   1   1   1   1   2   1   1   2   2   2   2   1
    1   1   1   1   2

 CLUSTER SIZES
    44   36

 CLUSTER  1 CONTAINS   44 DATA UNITS

 CENTROID COORDINATES
   0.7214E 03  0.2164E 06

 MEMBERSHIP LIST
    1   6   7   8  10  11  13  14  16  17  18  19  20  21  25  27  29  31  32  36  37  39  40  41  44
   45  46  47  50  52  54  57  58  64  65  66  67  69  70  75  76  77  78  79

 CLUSTER  2 CONTAINS   36 DATA UNITS

 CENTROID COORDINATES
   0.7400E 03  0.5739E 06

 MEMBERSHIP LIST
    2   3   4   5   9  12  15  22  23  24  26  28  30  33  34  35  38  42  43  48  49  51  53  55  56
   59  60  61  62  63  68  71  72  73  74  80
```

APPENDIX F

Coordinates of Data Points Used in This Study

INPUT DATA FOR EVENLY DISTRIBUTED CONTRIVED DATA - DATA1

i	x	y
1	10	20
2	10	45
3	10	60
4	10	71
5	15	52
6	15	35
7	20	15
8	20	23
9	22	59
10	22	17
11	25	11
12	25	70
13	27	33
14	30	20
15	30	46
16	32	12
17	32	23
18	35	15
19	38	35
20	40	10
21	39	20
22	40	52
23	40	65
24	40	76
25	45	15
26	45	43
27	50	30
28	50	56
29	52	34
30	55	73
31	55	23
32	55	12
33	61	44
34	60	50
35	62	64
36	65	15
37	67	22
38	68	57
39	70	35
40	75	10
41	75	25
42	75	47
43	75	65
44	77	12
45	80	15
46	85	30

i	x	y
47	83	39
48	85	52
49	82	60
50	90	20
51	90	65
52	96	28
53	94	46
54	100	36
55	100	55
56	100	70
57	105	15
58	110	18
59	108	41
60	110	60
61	110	72
62	112	68
63	115	52
64	115	24
65	115	20
66	120	15
67	120	27
68	122	47
69	125	17
70	125	22
71	123	74
72	125	65
73	130	55
74	127	43
75	130	29
76	130	15
77	135	7
78	140	20
79	143	33
80	138	46

INPUT DATA FOR UNEVENLY DISTRIBUTED CONTRIVED DATA - DATA2

i	x	y
1	5	65
2	7	67
3	10	55
4	10	65
5	10	70
6	10	75
7	15	60
8	15	65
9	15	77
10	15	80
11	20	55
12	20	65
13	20	70
14	22	58
15	25	45
16	25	55
17	25	65
18	25	75
19	30	50
20	35	45
21	40	50
22	45	45
23	45	50
24	45	55
25	47	52
26	50	40
27	50	45
28	50	55
29	50	60
30	53	48
31	55	60
32	55	55
33	55	45
34	55	40
35	57	36
36	60	33
37	60	50
38	62	45
39	63	41
40	63	36
41	65	38
42	67	41
43	55	28
44	59	28
45	59	23
46	64	22
47	62	20
48	58	15
49	55	20
50	50	10
51	45	10
52	40	15
53	45	20
54	75	25
55	85	30
56	90	25
57	90	35
58	100	35
59	100	40
60	105	30
61	105	45
62	105	50
63	110	28
64	110	36
65	110	43
66	115	32
67	115	40
68	115	45
69	115	48
70	115	55
71	119	36
72	120	43
73	122	51
74	125	47
75	125	40
76	130	33
77	133	25
78	128	27
79	125	30
80	125	25

INPUT DATA FOR NORTH BURNABY AREA (STREET LETTER BOX LOCATIONS)

ID	x	y	location
NB001	401.23	228.53	SFU MALL
NB002	398.23	229.73	SFU FRONT OF SHELL STATION
NB003	397.93	228.73	SFU FRONT OF STUDENT RES.
NB004	365.23	220.63	DUTHIE AND CURTIS
NB005	361.03	214.33	CLIFF AND WINCH
NB006	365.13	212.03	DUTHIE AND HALIFAX
NB007	365.63	211.63	1800 DUTHIE
NB008	369.93	209.33	PHILLIPS AND CORONADO
NB009	365.33	204.63	DUTHIE AND BROADWAY
NB010	372.13	204.83	CAMROSE AND BROADWAY
NB011	376.43	198.33	LAKE CITY AND ENTERPRISE
NB012	384.93	199.43	UNDERHILL AND ENTER
NB013	391.63	198.63	PRODUCTION WAY AND THUNDERBIRD CRES.
NB014	385.83	193.23	LAKEDALE AND GOVERNMENT
NB015	378.93	193.23	PIPER AND GOVERNMENT
NB016	373.93	193.23	LOZELLES AND GOVERNMENT
NB017	368.93	193.23	PHILLIPS AND GOVERNMENT
NB018	373.93	190.03	LOZELLES AND WINSTON
NB019	368.13	191.63	7342 WINSTON
NB020	363.43	202.03	BAINBRIDGE AND LOUGHEED
NB021	361.13	205.93	CLIFF AND BROADWAY
NB022	356.83	204.83	SPERLING AND BROADWAY
NB023	356.83	208.43	SPERLING AND ADAIR
NB024	356.83	216.53	SPERLING AND KITCHENER
NB025	356.83	220.83	SPERLING AND CURTIS
NB026	356.83	226.53	SPERLING AND HASTINGS (SUB LOCHDALE)
NB027	362.93	226.73	BARNET AND HASTINGS
NB028	364.33	228.83	BARNET AND PANDORA
NB029	364.73	232.33	INLET AND SIERRA
NB030	352.73	226.43	KENSINGTON AND HASTINGS
NB031	348.63	216.53	FELL AND KITCHENER
NB032	345.53	215.33	HOLDOM AND GRANT
NB033	348.53	211.43	FELL AND BUCHANAN
NB034	352.73	212.33	KENSINGTON AND HALIFAX
NB035	350.73	207.33	6265 EAST BROADWAY
NB036	346.73	208.93	5901 EAST BROADWAY(SUB 113)
NB037	344.43	209.93	HOLDOM AND BROADWAY (SUMAS)
NB038	332.23	210.43	BETA AND LOUGHEED
NB039	329.93	208.53	ALPHA AND DAWSON
NB040	327.03	210.83	4477 LOUGHEED
NB041	325.63	212.53	ROSSER AND HALIFAX
NB042	323.53	208.53	MADISON AND DAWSON
NB043	320.13	214.53	GILMORE AND GRAVELEY
NB044	313.33	213.83	3765 EAST 1ST
NB045	314.23	220.33	DOUGLAS AND ESMOND
NB046	318.03	226.03	MCDONALD AND PENDER (NB STAT.)

NB047	323.73	223.13	MADISON AND UNION
NB048	319.43	222.13	GILMORE AND VENABLES
NB049	319.43	220.03	GILMORE AND NAPIER
NB050	323.73	218.43	1265 MADISON (SUB 67)
NB051	323.73	214.93	MADISON AND GRAVELEY
NB052	327.73	214.93	WILLINGDON AND BRENTLAWN
NB053	330.73	216.03	FAIRLAWN AND MIDLAWN
NB054	336.13	219.03	DELTA AND FAIRLAWN
NB055	336.13	221.03	DELTA AND PARKER
NB056	342.53	223.13	HOWARD AND UNION
NB057	339.33	217.23	1381 SPRINGER
NB058	336.13	212.83	DELTA AND BRENTLAWN
NB059	327.73	210.83	WILLINGTON AND LOUGHEED
NB060	330.23	211.53	EATONS BRENTWOOD (SUB 121)
NB061	330.43	213.43	REAR BRENTWOOD SHOP CTR.
NB062	327.73	221.03	WILLINGTON AND PARKER
NB063	327.73	227.03	WILLINGTON AND HASTINGS (SUB 45)
NB064	332.03	227.03	BETA AND HASTINGS
NB065	336.13	226.13	DELTA AND HASTINGS
NB066	334.13	233.23	EMPIRE DR (GAMMA) AND CAMBRIDGE
NB067	336.13	233.23	DELTA AND CAMBRIDGE
NB068	337.43	231.03	HYTHE AND DUNDAS
NB069	341.43	231.03	GROSVENOR AND DUNDAS
NB070	344.53	231.03	HOLDOM AND DUNDAS
NB071	346.73	231.03	WARWICK AND DUNDAS
NB072	347.73	226.43	STRATFORD AND HASTINGS
NB073	343.53	226.43	ELLESMERE AND HASTINGS (SUB 132)
NB074	339.33	226.43	SPRINGER AND HASTINGS
NB075	332.03	223.13	BETA AND UNION
NB076	323.73	227.03	MADISON AND HASTINGS
NB077	323.73	230.73	MADISON AND TRIUMPH
NB078	319.43	227.03	GILMORE AND HASTINGS
NB079	319.43	230.73	GILMORE AND TRIUMPH
NB080	319.43	233.23	GILMORE AND CAMBRIDGE
NB081	319.43	236.33	GILMORE AND TRINITY
NB082	316.03	238.23	INGLETON AND EDINBURGH
NB083	314.33	235.83	ESMOND AND MCGILL
NB084	314.33	232.73	ESMOND AND OXFORD
NB085	312.53	230.73	BOUNDRY AND TRIUMPH
NB086	316.03	227.03	INGLETON AND HASTINGS
NB087	318.03	227.03	MACDONALD AND HASTINGS (SUB 106)

INPUT DATA FOR SOUTH BURNABY AREA (STREET LETTER BOX LOCATIONS)

ID	x	y	location
SB001	375.16	162.27	CANADA WAY AND WEDGEWOOD
SB002	372.36	165.67	CANADA WAY AND GOODLAD
SB003	367.66	171.27	CANADA WAY AND STANLEY
SB004	362.16	172.77	BUCKINGHAM AND BURRIS
SB005	357.66	178.97	CANADA WAY AND SPERLING
SB006	352.16	183.77	CANADA WAY AND LEDGER
SB007	349.86	185.57	4916 CANADA WAY
SB008	348.06	190.67	GODWIN AND SPROTT
SB009	344.86	187.17	MAHON AND SPRUCE
SB010	344.76	181.37	MAHON AND GILPIN
SB011	339.46	181.37	ROYAL OAK AND GILPIN
SB012	333.66	181.17	GARDEN GROVE AND MOSCROP
SB013	342.16	189.07	5325 KINCAID
SB014	345.16	191.57	DOUGLAS AND WOODSWORTH
SB015	338.03	206.93	2210 DOUGLAS ROAD
SB016	344.06	200.17	DOUGLAS AND REGENT
SB017	339.66	198.57	ROYAL OAK AND MANOR
SB018	339.66	194.57	4694 CANADA WAY
SB019	333.86	196.37	GARDNER COURT AND CANADA WAY
SB020	328.66	191.47	3700 WILLINGTON (FRONT BCIT)
SB021	330.26	190.17	3700 WILLINGTON (SAC BCIT)
SB022	328.06	196.17	WILLINGDON AND CANADA WAY
SB023	324.36	196.17	SUMNER AND CANADA WAY
SB024	324.36	197.97	SUMNER AND DOMINION
SB025	326.36	203.37	GILMORE AND STILL CREEK
SB026	316.36	202.17	SMITH AND MYRTLE
SB027	315.16	196.17	3737 CANADA WAY
SB028	321.21	192.07	KALYK AND NITHSDALE
SB029	316.86	190.67	3815 SUNSET (SUB 93)
SB030	317.96	190.67	BURNABY GEN HOSPITAL
SB031	316.16	187.97	SMITH AND SPRUCE
SB032	316.06	182.87	SMITH AND MOSCROP
SB033	320.86	182.87	PATTERSON AND MOSCROP
SB034	320.86	178.17	PATTERSON AND HAZELWOOD
SB035	322.56	174.57	BARKER AND BOND
SB036	327.06	176.17	GILPIN CRES AND BURKE
SB037	324.66	179.62	BARKER AND GILPIN
SB038	324.86	182.87	DARWIN AND MOSCROP
SB039	328.76	167.77	MCKAY AND KINGSWAY
SB040	326.26	164.07	MCKAY AND BERESFORD
SB041	321.36	167.57	PATTERSON AND BERESFORD
SB042	323.46	161.47	CASSIE AND MAYWOOD
SB043	328.36	162.77	BERESFORD AND TELFORD
SB044	326.46	159.17	SUSSEX AND IMPERIAL
SB045	324.26	155.07	MCKAY AND VICTORY
SB046	321.36	151.17	PATTERSON AND RUMBLE

SB047	321.36	148.07	PATTERSON AND PORTLAND
SB048	326.46	143.17	SUSSEX AND PORTLAND
SB049	330.36	146.77	STRATHEARN AND MCKEE
SB050	326.46	151.07	SUSSEX AND RUMBLE
SB051	329.76	153.12	FREDERICK AND WATLING
SB052	335.36	151.07	NELSON AND RUMBLE
SB053	335.36	146.77	NELSON AND MCKEE
SB054	335.36	155.07	NELSON AND VICTORY
SB055	336.36	159.17	DUNBLANE AND IMPERIAL
SB056	339.61	156.07	ROYAL OAK AND BERESFORD
SB057	343.66	155.17	MCPHERSON AND BERESFORD
SB058	347.86	153.77	BULLER AND BERESFORD
SB059	347.86	150.97	BULLER AND RUMBLE
SB060	347.86	147.97	BULLER AND PORTLAND
SB061	347.86	144.87	BULLER AND CARSON
SB062	347.56	140.47	GILLEY AND MARINE
SB063	352.16	147.87	GILLEY AND PORTLAND
SB064	352.16	151.92	7542 GILLEY
SB065	343.66	150.97	MCPHERSON AND RUMBLE
SB066	343.66	146.87	MCPHERSON AND MCKEE
SB067	339.86	150.97	ROYAL OAK AND RUMBLE (SUB 92)
SB068	339.86	148.97	ROYAL OAK AND CLINTON
SB069	339.86	145.97	ROYAL OAK AND EWART
SB070	339.86	141.97	ROYAL OAK AND MARINE
SB071	326.96	144.92	SUSSEX AND MARINE
SB072	321.36	144.97	PATTERSON AND MARINE
SB073	317.06	143.77	GREENALL AND MARINE
SB074	315.46	146.77	JOFFRE AND CARSON
SB075	314.86	151.12	JOFFRE AND RUMBLE (SUB 109)
SB076	314.86	154.07	JOFFRE AND ARBOR
SB077	314.86	158.07	JOFFRE AND DUBOIS
SB078	332.76	176.17	SUSSEX AND BURKE
SB079	332.76	171.92	SUSSEX AND SARDIS
SB080	335.26	173.07	NELSON AND BUXTON
SB081	339.36	168.02	ROYAL OAK AND DOVER
SB082	344.21	163.77	ELGIN AND IRVING
SB083	348.81	165.57	WALTHAM AND BERWICK
SB034	352.06	163.57	GILLEY AND BURNS
SB085	355.26	164.97	BRANTFORD AND STANLEY
SB086	357.56	168.37	SPERLING AND WALKER
SB087	357.56	161.47	SPERLING AND BURFORD
SB088	360.26	165.07	WALKER AND STANLEY (SUB 97)
SB089	363.26	161.37	WALKER AND IMPERIAL
SB090	372.16	159.77	MARY AND VISTA
SB091	369.76	160.77	HUMPHRIES AND ELWELL
SB092	366.56	159.07	LINDEN AND ELWELL
SB093	367.76	155.07	ESMONDS AND KINGSWAY
SB094	364.46	155.27	7155 KINGSWAY (SUB 120)
SB095	363.26	155.17	SALISBURY AND KINGSWAY
SB096	352.96	159.17	COLBOURNE AND IMPERIAL

SB097	357.56	155.17	SPERLING AND KINGSWAY
SB098	352.06	156.87	GILLEY AND KINGSWAY
SB099	346.36	159.17	KINGSWAY AND IMPERIAL
SB100	342.76	160.82	DENBIGH AND KINGSWAY
SB101	339.36	162.57	ROYAL OAK AND KINGSWAY (SUB 85)
SB102	333.26	159.17	JUBILEE AND IMPERIAL
SB103	330.76	164.42	SIMPSON SEARS (SUB 65)
SB104	332.26	164.37	SIMPSON SEARS
SB105	335.26	164.52	NELSON AND KINGSWAY
SB106	333.76	165.37	MCMURRAY AND KINGSWAY
SB107	331.36	166.47	SUSSEX AND KINGSWAY
SB108	331.86	167.47	6025 SUSSEX
SB109	329.51	169.37	PIONEER AND GRANGE
SB110	326.96	168.87	4429 KINGSWAY (SUB 122)
SB111	323.50	169.67	4211 KINGSWAY
SB112	318.11	171.47	JERSEY AND KINGSWAY (SUB 88)
SB113	316.06	173.57	SMITH AND THURSTON

$SIG

APPENDIX G

Membership Lists of Groupings Defined by Various Clustering Methods

MEMBERSHIP LIST OF GROUPS DEFINED BY VARIOUS METHODS
EVENLY DISTRIBUTED CONTRIVED DATA SET - DATA1

SINGLE LINKAGE : EUCLIDEAN DISTANCE + SUM OF DIFFERENCES METHODS

GROUP 1 - 67 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55	56	59	60	61	62
63	68	71	72	73	74	80								

GROUP 2 - 13 MEMBERS

57	58	64	65	66	67	69	70	75	76	77	78	79

SINGLE LINKAGE : CHI-SQUARES METHOD

GROUP 1 - 22 MEMBERS

1	2	3	4	5	6	8	9	12	13	15	19	22	23	24
26	28	30	34	35	38	43								

GROUP 2 - 58 MEMBERS

7	10	11	14	16	17	18	20	21	25	27	29	31	32	33
36	37	39	40	41	42	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
68	69	70	71	72	73	74	75	76	77	78	79	80		

COMPLETE LINKAGE METHOD

GROUP 1 - 35 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
33	34	35	38	42										

GROUP 2 - 45 MEMBERS

31	32	36	37	39	40	41	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

AVG. LINKAGE BETWEEN MERGED GROUP

GROUP 1 - 51 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51									

GROUP 2 - 29 MEMBERS

52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
67	68	69	70	71	72	73	74	75	76	77	78	79	80	

AVG. LINKAGE WITHIN NEW GROUP

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	53	55							

GROUP 1 - 53 MEMBERS
GROUP 2 - 27 MEMBERS

52	54	56	57	58	59	60	61	62	63	64	65	66	67	68
69	70	71	72	73	74	75	76	77	78	79	80			

CENTROID AND MEDIAN METHODS
GROUP 1 - 49 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
48	49	50	51											

GROUP 2 - 31 MEMBERS

46	47	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80														

WARD'S METHOD
GROUP 1 - 43 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	36	37	40	41	44	45	46	50	52		

GROUP 2 - 37 MEMBERS

35	38	39	42	43	47	48	49	51	52	53	54	55	56	57
58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80							

JANCEY'S, FORGY'S AND CONVERGENT K-MEAN METHODS
GROUP 1 - 36 MEMBERS

2	3	4	5	9	12	15	22	23	24	26	28	30	33	34
35	38	42	43	48	49	51	53	55	56	59	60	61	62	63
68	71	72	73	74	80									

GROUP 2 - 44 MEMBERS

1	6	7	8	10	11	13	14	16	17	18	19	20	21	25
27	29	31	32	36	37	39	40	41	44	45	46	47	50	52
54	57	58	64	65	66	67	69	70	75	76	77	78	79	

MEMBERSHIP LIST OF GROUPS DEFINED BY VARIOUS METHODS
UNEVENLY DISTRIBUTED CONTRIVED DATA SET - DATA2

SINGLE LINKAGE : EUCLIDEAN DISTANCE METHOD
MEMBERSHIP LIST NOT ANALYSED

SINGLE LINKAGE : SUM OF DIFFERENCES AND CHI-SQUARES METHODS
GROUP 1 - 17 MEMBERS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 16
17 18
GROUP 2 - 25 MEMBERS
15 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42
GROUP 3 - 38 MEMBERS
43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80

COMPLETE LINKAGE METHOD
GROUP 1 - 19 MEMBERS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19
GROUP 2 - 34 MEMBERS
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
50 51 52 53
GROUP 3 - 27 MEMBERS
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
69 70 71 72 73 74 75 76 77 78 79 80

AVG. LINKAGE METHODS
GROUP 1 - 20 MEMEBERS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20
GROUP 2 -33 MEMBERS
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53
GROUP 3 - 27 MEMBERS
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
69 70 71 72 73 74 75 76 77 78 79 80

CENTROID AND MEDIAN METHODS

GROUP 1 - 20 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20										

GROUP 2 - 37 MEMBERS

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57								

GROUP 3 - 23 MEMBERS

58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80							

WARD'S METHOD

GROUP 1 - 42 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42			

GROUP 2 - 14 MEMBERS

43	44	45	46	47	48	49	50	51	52	53	54	55	56

GROUP 3 - 24 MEMBERS

57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79	80						

JANCEY'S METHOD

GROUP 1 - 23 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	16
17	18	24	28	29	31	32	70							

GROUP 2 - 23 MEMBERS

36	43	44	45	46	47	48	49	50	51	52	53	54	55	56
60	63	66	76	77	78	79	80							

GROUP 3 - 34 MEMBERS

15	19	20	21	22	23	25	26	27	30	33	34	35	37	38
39	40	41	42	57	58	59	61	62	64	65	67	68	69	71
72	73	74	75											

FORGY'S AND CONVERGENT K-MEAN METHODS

GROUP 1 - 16 MEMBERS

1	2	4	5	6	7	8	9	10	12	13	14	17	18	29
31														

GROUP 2 - 29 MEMBERS

35	36	40	43	44	45	46	47	48	49	50	51	52	53	54
55	56	57	58	60	63	64	66	71	76	77	78	79	80	

GROUP 3 - 35 MEMBERS

3	11	15	16	19	20	21	22	23	24	25	26	27	28	30
32	33	34	37	38	39	41	42	59	61	62	65	67	68	69
70	72	73	74	75										

MEMBERSHIP LIST OF GROUPS DEFINED BY VARIOUS METHODS
EMPIRICAL DATA - NORTH BURNABY AREA - NBDATA

SINGLE LINKAGE: EUCLIDEAN DISTANCE AND SUM OF DIFFERENCES METHODS +
AVG. LINKAGE BETWEEN MERGED GROUP METHOD

GROUP 1 - 3 MEMBERS

1	2	3

GROUP 2 - 84 MEMBERS

4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87						

SINGLE LINKAGE : CHI-SQUARES METHOD

GROUP 1 - 29 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	33	34	35	36	37	

GROUP 2 - 58 MEMBERS

25	26	27	28	29	30	31	32	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87		

COMPLETE LINKAGE METHOD

GROUP 1 - 37 MEMBERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37								

GROUP 2 - 50 MEMBERS

38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
68	69	70	71	72	73	74	75	76	77	78	79	80	81	82
83	84	85	86	87										

AVG. LINKAGE WITHIN NEW GROUP METHOD

GROUP 1 - 39 MEMBERS

1	2	3	38	39	40	41	42	43	44	45	46	47	48	49
50	51	52	53	58	59	60	61	62	63	64	75	76	77	78
79	80	81	82	83	84	85	86	87						

GROUP 2 - 48 MEMBERS

4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	54	55	56	57	65	66	67	68	69	70	71
72	73	74												

CENTROID AND MEDIAN METHODS

GROUP 1 - 12 MEMBERS

1 2 3 11 12 13 14 15 16 17 18 19

GROUP 2 - 75 MEMBERS

4 5 6 7 8 9 10 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81 82 83 84 85 86 87

WARD'S METHOD

GROUP 1 - 35 MEMBERS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 31 32 33 34 35
36 37 54 55 57

GROUP 2 - 52 MEMBERS

26 27 28 29 30 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53 56 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87

JANCEY'S METHOD

GROUP 1 - 35 MEMBERS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 33 34 35 36

GROUP 2 - 52 MEMBERS

32 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87

FORGY'S AND CONVERGENT K-MEAN METHODS

GROUP 1 - 31 MEMBERS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 34
35

GROUP 2 - 56 MEMBERS

30 31 32 33 36 37 38 39 40 41 42 43 44 45 46
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61
62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
77 78 79 80 81 82 83 84 85 86 87

MEMBERSHIP LIST OF GROUPS DEFINED BY VARIOUS METHODS
EMPIRICAL DATA - SOUTH BURNABY AREA - SBDATA

SINGLE LINKAGE : SUM OF DIFFERENCES METHOD +
AVG. LINKAGE BETWEEN MERGED GROUP METHOD

GROUP 1 - 21 MEMBERS

1	2	3	4	5	83	84	85	86	87	88	89	90	91	92
93	94	95	96	97	98									

GROUP 2 - 26 MEMBERS

6	7	8	9	10	11	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30	31	32				

GROUP 3 - 66 MEMBERS

12	33	34	35	36	37	38	39	40	41	42	43	44	45	46
47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
62	63	64	65	66	67	68	69	70	71	72	73	74	75	76
77	78	79	80	81	82	99	100	101	102	103	104	105	106	107
108	109	110	111	112	113									

SINGLE LINKAGE : CHI SQUARES METHOD

GROUP 1 - 34 MEMBERS

1	2	48	49	52	53	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	87	89	90	91	92	93	94
95	96	97	98											

GROUP 2 - 36 MEMBERS

3	4	5	40	42	43	44	45	46	47	50	51	54	55	56
74	75	76	77	81	82	83	84	85	86	88	99	100	101	102
103	104	105	106	107	108									

GROUP 3 - 43 MEMBERS

6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
36	37	38	39	41	78	79	80	109	110	111	112	113		

COMPLETE LINKAGE METHOD

GROUP 1 - 20 MEMBERS

1	2	3	4	83	84	85	86	87	88	89	90	91	92	93
94	95	96	97	98										

GROUP 2 - 45 MEMBERS

45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	81	82	99	100	101	102	103	104	105	106	107	108

GROUP 3 - 48 MEMBERS

5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
35	36	37	38	39	40	41	42	43	44	78	79	80	109	110
111	112	113												

AVG. LINKAGE WITHIN NEW GROUP METHOD
GROUP 1 - 22 MEMBERS
1 2 3 4 5 83 84 85 86 87 88 89 90 91 92
93 94 95 96 97 98 99
GROUP 2 - 36 MEMBERS
46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 82 100 101 102
GROUP 3 - 55 MEMBERS
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 78 79 80 81 103
104 105 106 107 108 109 110 111 112 113

CENTROID AND MEDIAN METHODS
GROUP 1 - 2 MEMBERS
62 70
GROUP 2 - 18 MEMBERS
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31 32
GROUP 3 - 93 MEMBERS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 33
34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 63 64
65 66 67 68 69 71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110
111 112 113

WARD'S METHOD
GROUP 1 - 45 MEMBERS
1 2 3 4 52 53 54 55 56 57 58 59 60 61 62
63 64 65 66 67 68 69 70 81 82 83 84 85 86 87
88 89 90 91 92 93 94 95 96 97 98 99 100 101 102
GROUP 2 - 28 MEMBERS
5 6 7 8 9 10 11 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33
GROUP 3 - 40 MEMBERS
12 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 71 72 73 74 75 76 77 78 79 80 103
104 105 106 107 108 109 110 111 112 113

JANCEY'S METHOD
GROUP 1 - 33 MEMBERS

6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	36
37	38	78												

GROUP 2 - 46 MEMBERS

35	39	40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	67	68	69	70	71	72	73	74	75	76	77
79	80	81	101	102	103	104	105	106	107	108	109	110	111	112
113														

GROUP 3 - 34 MEMBERS

1	2	3	4	5	57	58	59	60	61	62	63	64	65	66
82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100											

FORGY'S AND CONVERGENT K-MEAN METHODS
GROUP 1 - 34 MEMBERS

6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
36	37	38	78											

GROUP 2 - 49 MEMBERS

39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
54	55	56	57	65	66	67	68	69	70	71	72	73	74	75
76	77	79	80	81	100	101	102	103	104	105	106	107	108	109
110	111	112	113											

GROUP 3 - 30 MEMBERS

1	2	3	4	5	58	59	60	61	62	63	64	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

$SIG

APPENDIX H

Listing and Sample Outputs from ROUTE: a Computer Program Designed for Evaluating Clusters

```
      DIMENSION X(100),Y(100),D(100,100),S(100,100),DDJ(100),
     1          IT(100),NI(5000),NJ(5000),L(100),SV(100,10),
     2          LR(50,100),LLR(10,100),LK(100),PROTIT(15)
     3          ,FSA(2),TIME(100),LAB(100),DISTZ(101,101),
     4         AREA(5),NLR(100),SOT(3,5000),SNAME(100,10),
     5       LAB1(100),TITLE(20),WAY(20)
C
C      THE FOLLOWING ARE IMPORTANT VARIABLES:-
C
C           PROTIT    = PROBLEM TITLE
C           AREA(I)   = GEOGRAPHIC LOCATION
C           ICAP      = CAPACITY OF TRUCK (NO. OF STOPS)
C           ITIM      = TIME CONSTRAINT OF TRUCK ROUTE
C           M         = TOTAL NUMBER OF TRUCKS
C           INUM      = TOTAL NUMBER OF BOXES IN THIS AREA
C           ISTIME    = STOPPING TIME AT EACH BOX
C           AVGSPD    = AVERAGE TRUCK SPEED IN M.P.H.
C           SCALE     = MAP SCALE (1 MM= XX  MILES)
C           NPLOT     = PLOTTING CONTROL NUMBER
C                       NPLOT=1      EXISTING ROUTE PLOTTED
C                       NPLOT.GT.1   ONLY NEW ROUTE PLOTTED
C           XORD      = X-COORDINATE OF STATION IN MM
C           YORD      = Y-COORDINATE OF STATION IN MM
C           N         = NO OF BOXES FOR THIS ROUTE
C           FSA       = FORWARD SORTATION AREA
C           TYPE      = SLB/BR
C           XORG      = X-COORDINATE FOR PLOT ORIGIN (INCHES)
C           YORG      = Y-COORDINATE FOR PLOT ORIGIN (INCHES)
C
      READ (5,11) PROTIT
   11 FORMAT (15A4)
      WRITE (6,190) PROTIT
  190 FORMAT ('1',20(/),25X,'MULTIPLE ROUTE SCHEDULING BY SAVINGS ',
     1'ALGORITHM---       ',/26X,15A4)
      READ(5,15)(AREA(I),I=1,5),ICAP,ITIM,M,INUM,ISTIME,NPLOT,AVGSPD,
     A           SCALE
15    FORMAT(5A4,6I4,2F10.6)
      WRITE(6,60)(AREA(I),I=1,5),ICAP,ITIM,AVGSPD,M,INUM,SCALE
 60   FORMAT('1',10(/),15X,'THE FOLLOWING ROUTING PRINT-OUTS FOR ',
     1      5A4,' AREA ARE BASED ON : -'//,17X,
     2      'CAPACITY CONSTRAINT OF EACH TRUCK ROUTE = ',I6,2X,'BOXES'//
     3,17X,'TIME CONSTRAINT OF EACH ROUTE',11X,'= ',I6,2X,'MINUTES'//,
     4  17X,'AVERAGE SPEED OF TRUCKS EN ROUTE',8X,'= ',F6.2,2X,'M.P.H.'/
     5/,17X,'TOTAL NUMBER OF ROUTES IN THE AREA',6X,'= ',I6,2X,'ROUTES'/
     6/,17X,'TOTAL NUMBER OF BOXES IN THE AREA',7X,'= ',I6,2X,'BOXES'/
     7/,17X,'MAP SCALE',31X,'= ',2X,'1 MM =',F10.6,'MILES',/)
      WRITE(6,61) ISTIME
 61   FORMAT(17X,'STOPPING TIME FOR RESIDENTIAL BOXES',5X,'=',5X,I2,2X,
     1'SECONDS',//,17X,'STOPPING TIME FOR BUSINESS BOXES',8X,'=',5X,
     2'AS SPECIFIED',//)
```

```
      JC=0
      IR=0
      TDIST=0.
      READ(5,18) FSA,TYPE,XORD,YORD,XORG,YORG,IPLOT,ILIST
18    FORMAT(5X,3A4,4F8.2,2I4)
7     DO 987 IL=1,100
987   IT(IL)=2
C          X(.)      - X-COORDINATE OF CALL POINT (.)
C          Y(.)      - Y-COORDINATE OF CALL POINT (.)
C          L(.)      - LABEL OF CALL POINT (.)
C          TIME(.)   - STOPPING TIME REQUIRED AT POINT (.)
C          D(.,.)    - DISTANCE BETWEEN CALL POINTS (.,.)
C          DDO(.)    - DISTANCE OF CALL POINT (.) FROM (0,0)
C          S(.,.)    - SAVINGS BY JOINING CALL POINTS(.,.)
C          LR(I,J)    - LABEL OF JTH ELT. IN SUBROUTE I
C          NLR(.)     - NO. OF ELT. IN SUBROUTE(.)
C          NR         - NO. OF CURRENT SUBROUTES
C
C
C      READ LABEL AND LOCATION OF EACH BOX
C
      CALL SELECT(L,X,Y,TIME,SNAME,INUM,TITLE,WAY,N)
C
C      CALCULATE DISTANCE BETWEEN BOX AND ORIGIN
C
 92   DO 1 I=1,N
1     DDO(I)=(ABS(X(I)-XORD)+ABS(Y(I)-YORD))
      NM1=N-1
C
C      CALCULATE DISTANCE SAVED MATRIX AND DISTANCE MATRIX
C D=DISTANCE MATRIX  S=DISTANCE SAVED
      DO 3 I=1,NM1
      K=I+1
      DO 3 J=K,N
      D(I,J)=(ABS(X(I)-X(J)))+(ABS(Y(I)-Y(J)))
      S(I,J)=DDO(I)+DDO(J)-D(I,J)
      S(J,I)=S(I,J)
      D(J,I)=D(I,J)
    3 CONTINUE
      NA=N+1
      DO 5 I=1,NA
      D(I,I)=0.
 5    S(I,I)=0.
      K=0
      CALL COUNT(D,N,TITLE,WAY,SCALE)
C
C      STORE DISTANCE SAVED MATRIX IN A VECTOR
C
C  SOT(1,.)=FROM-MAILBOX,SOT(2,.)=TO-MAILBOX,SOT(3,.)=DIST-SAVED
      DO 4 I=1,NM1
```

```
      IP1=I+1
      DO 4 J=IP1,N
      K=K+1
      SOT(1,K)=I
      SOT(2,K)=J
      SOT(3,K)=S(I,J)
    4 CONTINUE
C
C      SORT VECTOR BY DISTANCE SAVED VALUES
C
      NN=((N*N)-N)/2
      CALL ISORT(SOT,3,3000,1,NN,3,3,-1)
      DO 10 I=1,NN
      NI(I)=SOT(1,I)
      NJ(I)=SOT(2,I)
   10 CONTINUE
C
C             BEGIN ROUTE CHOICE HEURISTIC.
C
C             SUBROUTES ARE FORMED BY SEARCHING IN THE LIST SOT(3,.) FOR
C             MAX. SAVINGS AND ARE LINKED MULTIPLLY.
C
      NR=0
   20 DO 12 I=1,NN
C
C      IF NEGATIVE, IT IS ALREADY IN ROUTE
C
      IF (SOT(3,I).LT.0.) GO TO 12
C
C      TEST TO SEE IF ONLY ONE IS END POINT OF SUBROUTE
C
      IF (IT(NJ(I)).NE.IT(NI(I))) GO TO 21
C
C
C      TEST TO SEE IF IN SUBROUTE
C
      IF (IT(NJ(I)).EQ.1) GO TO 22
C
C      BEGIN NEW SUBROUTE
C
      IT(NI(I))=1
      IT(NJ(I))=1
C NR=# OF NEW ROUTES,LR(NR,1)=LABEL.1ST.BOX,LR'NR,2)=LABEL.2.BOX
C NLR(NR)=2.NO.OF.LABELS=2
      NR=NR+1
      LR(NR,1)=NI(I)
      LR(NR,2)=NJ(I)
      NLR(NR)=2
      GO TO 12
C
```

```
C      ADD NI(I) TO SUBROUTE
C
22    DO 71 J=1,NR
C EG ROUT1=12-14-17 ROUT2=18-20-12 TRY TO MATCH AND MAKE 1 ROUTE
C 1 ROUTE=18-2--12-14-17
C      FIND SUBROUTE(S) AND END
C CHECK TO SEE IF EITHER HAS BEEN JOINED TOGETHER BEFORE SO THAT
C A CLOSED LOOP IS NOT COMPLETED TOO EARLY
      IF (LR(J,1).EQ.NI(I).OR.LR(J,NLR(J)).EQ.NI(I)) K1=J
      IF (LR(J,1).EQ.NJ(I).OR.LR(J,NLR(J)).EQ.NJ(I)) K2=J
   71 CONTINUE
      IF (K1.EQ.K2) GO TO 12
C
C      TEST FOR BOTH AT BEGINNING OF SUBROUTE
C IF NEITHER LABEL OCCURRED BEFORE STORE IN NEW ROW OF MATRIX BUT
C DO NOT CHECK FOR ANYTHING
      IF (LR(K1,1).EQ.NI(I).AND.LR(K2,1).EQ.NJ(I)) GO TO 73
C
C      TEST FOR ROUTES FITTING TOGETHER
C
      IF (LR(K1,NLR(K1)).EQ.NI(I).AND.LR(K2,NLR(K2)).EQ.NJ(I)) GO TO 79
C IF OLD BOX LABEL=NEWBOX LABEL GO TO 72
      IF (LR(K1,NLR(K1)).EQ.NI(I)) GO TO 72
      GO TO 99
C
C      REVERSE ORDER OF SUBROUTE K1 & PLACE INTO LLR
C
   73 NK1=NLR(K1)
      DO 75 J=1,NK1
      LLR(K1,NLR(K1)-J+1)=LR(K1,J)
   75 CONTINUE
C STORE NEW LABEL IN NEW ROW OF MATRIX
      DO 88 J=1,NK1
      LR(K1,J)=LLR(K1,J)
   88 CONTINUE
C
C      PLACE BOTH SUBROUTES INTO ROUTE K1
C  NK1P2=SUM.TOTAL.NO.LABELS   NK1=NO.OF.BOXES.IN.K1.ROUTE
   72 NK1P2=NLR(K1)+NLR(K2)
      NK1=NLR(K1)+1
C JOIN UP 2 SUBROUTES TOGETHER
      DO 76 J=NK1,NK1P2
      LR(K1,J)=LR(K2,J-NLR(K1))
      LR(K2,J-NLR(K1))=0
   76 CONTINUE
      NLR(K1)=NK1P2
C
C      TURN LABELS AROUND
C
      IF(K2.GT.K1) GO TO 78
```

```
   89 K2P1=K2+1
C INCORPORATE NEW ROUTE AND RENUMBER OLD ROUTE SO OLD ONE FITS
C NLR=# OF BOXES IN ROUTE COUNTER   LR=NEW LABELS
      DO 77 J=K2P1,NR
      NLRJ=NLR(J)
      NLR(J-1)=NLR(J)
      DO 77 K=1,NLRJ
      LR(J-1,K)=LR(J,K)
   77 CONTINUE
      NR=NR-1
      GO TO 100
78    IF(K2.EQ.NR) GO TO 67
      NNR=NR-1
      DO 66 K=K2,NNR
      JAK=K+1
      NNUM=NLR(JAK)
      DO 66 J=1,NNUM
      LR(K,J)=LR(JAK,J)
      LR(JAK,J)=0
      NDUMP=NLR(K)
      NLR(K)=NLR(JAK)
66    CONTINUE
      NLR(NR)=0
67    NR=NR-1
      GO TO 100
C
C      FIT SECOND ROUTE INTO LLR
C
   79 NK2=NLR(K2)
      DO 81 J=1,NK2
      LLR(K2,NLR(K2)-J+1)=LR(K2,J)
   81 CONTINUE
      DO 80 J=1,NK2
      LR(K2,J)=LLR(K2,J)
   80 CONTINUE
      GO TO 72
   99 KK1=K1
      K1=K2
      K2=KK1
      GO TO 72
C
C      PLACE NEW LOCATIONS ON SUBROUTE
C
21    IF (IT(NI(I)).EQ.2) GO TO 52
C SWITCH LABELS FOR NEW BOXES THE OLD BOX HAS ALREADY JOINED WITH
C ANOTHER NOW IT BECOMES A "TO= BOX RATHER THAN A "FROM" BOX
      NII=NI(I)
      NI(I)=NJ(I)
      NJ(I)=NII
   52 L1=0
```

```
      L2=0
      DO 53 J=1,NR
C IF LABELS MATCH L1=OLD.LABEL CHECK 2ND LABEL WITH NEW LABEL
      IF (LR(J,1).EQ.NJ(I)) L1=J
      IF (LR(J,NLR(J)).EQ.NJ(I)) L2=J
   53 CONTINUE
      IF (L1.GT.0.) GO TO 55
      NLR(L2)=NLR(L2)+1
      LR(L2,NLR(L2))=NI(I)
      GO TO 100
   55 NL1=NLR(L1)
C RELABEL NEW BOX TO OLD 1ST BOX.AND RELABEL 1ST OLD BOX TO 2ND
C BOX NO.
      DO 56 J=1,NL1
      LR(L1,NL1-J+2)=LR(L1,NL1-J+1)
   56 CONTINUE
      LR(L1,1)=NI(I)
      NLR(L1)=NLR(L1)+1
C
C      CALCULATE STATISTICS AND UPDATE RECORDS
C
  100 IP1=I+1
C NLR=# OF BOXES JOINED TOGETHER
      IT(NI(I))=IT(NI(I))-1
      IT(NJ(I))=IT(NJ(I))-1
      IF (IT(NI(I)).EQ.0) GO TO 120
C GOTO ELINIMATE THE POSS OF JOINING AN OLD BOX
C GOTO ELINIMATE THE POSS OF JOINING AN OLD BOX
  135 IF (IT(NJ(I)).EQ.0) GO TO 128
      GO TO 12
C ELIMINATE POSS OF JOINING OLD BOXES BY MAKING NEGATIVE
  120 DO 125 J=IP1,NN
      IF (NI(J).EQ.NI(I)) SOT(3,J)=-1.
      IF (NJ(J).EQ.NI(I)) SOT(3,J)=-1.
  125 CONTINUE
      GO TO 135
  128 DO 126 J=IP1,NN
      IF (NI(J).EQ.NJ(I)) SOT(3,J)=-1.
      IF (NJ(J).EQ.NJ(I)) SOT(3,J)=-1.
  126 CONTINUE
   12 CONTINUE
C
C            PRINT RESULTS.
C
      JC=JC+1
      IR=IR+1
      DIST=0.
      IFLAG=1
      DO 35 I=1,NM1
   35 DIST=DIST+D(LR(1,I),LR(1,I+1))
```

```
C DIST=THE DISTANCE IS SUMMED FROM 1ST BOX TO LAST BOX
C DISS=THE COMPLETE ROUTE DISTANCE,BOTH ARE NECESSARY CAUSE
C THE ROUTE CAN BE STARTED  AFTER SLB OR BR OR IN MIDDLE OF SLB
      DISS=DIST+DDO(LR(1,1))+DDO(LR(1,N))
C      ADIST  = ACTUAL DISTANCE (STN TO STN,IN MILES)
C      BDIST  = ACTUAL DISTANCE (1ST TO LAST BOX, IN MILES)
C      STIME  = TOTAL TIME REQUIRED FOR STOPPING AT BOXES
C      TRTIME = TOTAL TRAVEL TIME REQUIRED (STN TO STN)
C      BTTIME = TOTAL TRAVEL TIME (1ST TO LAST BOX)
C      TOTIME = TOTAL TIME PER ROUTE INCLUDING STIME AND TRTIME
C                (STN TO STN)
C      BTIME  = TOTAL TIME FOR TRAVELLING AND STOPPING (1ST TO LAST BOX)
C      DIST   = DISTANCE OF THE ROUTE FROM FIRST BOX TO LAST BOX
C      DISS   = DISTANCE OF THE ROUTE FROM ORIGIN (STATION) TO ORIGIN
C
C CHANGE LR,LAK,ETC TO LAB OR NEMONICS FOR LABEL
C
      DO 34 I=1,N
 34   LAB(I)=LR(1,I)
      LAB(N+1)=0
 44   STIME=0.0
      DO 32 I=1,N
      LK(I)=L(LAB(I))
 32   CONTINUE
      DO 36 IN=1,N
      DO 36 INO=1,10
 36   SN(IN,INO)=SNAME(LAB(IN),INO)
      DO 33 IK=1,N
 33   STIME=(STIME+TIME(LR(1,IK)))
      BDIST=DIST*SCALE
      BTTIME=(BDIST/AVGSPD)*60.
      ADIST=DISS*SCALE
      TRTIME=(ADIST/AVGSPD)*60.
      TOTIME=TRTIME+STIME
      BTIME=BTTIME+STIME
      WRITE(6,41) IR,(AREA(IJ1),IJ1=1,5),(FSA(IJ2),IJ2=1,2),TYPE
 41   FORMAT('1',//,110X,'PAGE',I3,//,15X,'MULTIPLE ROUTE SCHEDULING',
     1       /,15X,'USING TIME-SAVING METHOD',///,15X,5A4,' AREA',/,
     2       15X,'F.S.A.',2X,2A4,/,15X,A4,' ROUTES',////)
      IF(IFLAG.EQ.2) GO TO 45
      WRITE(6,40)
 40   FORMAT(15X,'** PRELIMINARY ROUTE **',////)
      GO TO 46
 45   WRITE(6,47)
 47   FORMAT(15X,'** IMPROVED ROUTE **',////)
 46   WRITE(6,42) JC,N,STIME,BDIST,BTTIME,BTIME,ADIST,TRTIME,TOTIME
 42   FORMAT(15X,'ROUTE NO.',I4,//,
     115X,'NO. OF BOXES EN ROUTE',16X,'=',I5,3X,'BOXES',//,
     215X,'TOTAL STOPPING TIME',18X,'=',3X,F6.2,'  MINUTES',//,
     315X,'DISTANCE TRAVELLED(1ST TO LAST BOX)  =',3X,F6.2,'  MILES',//,
```

```
    415X,'TRAVEL TIME REQUIRED(1ST TO LAST BOX)=',3X,F6.2,'  MINUTES',/
    5/,15X,'TOTAL TIME REQUIRED(1ST TO LAST BOX) =',3X,F6.2,2X,'MINUTES
    6',//,15X,'TOTAL DISTANCE TRAVELLED(STN TO STN) =',3X,F6.2,
    7'  MILES',//,15X,'TOTAL TRAVEL TIME(STN TO STN)',8X,'=',
    83X,F6.2,'  MINUTES',//,15X,'TOTAL TIME REQUIRED FOR THIS ',
    9'ROUTE   =',3X,F6.2,2X,'MINUTES',/)
     NP1=N+1
     LR(1,NP1)=0
     LK(NP1)=0
     WRITE(6,43)(LK(J),J=1,NP1)
 43  FORMAT(///,15X,'ORDER OF CALL POINTS:',/,15X,'STATION - ',
    1       12(I5,'-'),/,6(25X,12(I5,'-'),/),' STATION',/)
     IF(IFLAG.EQ.2) GO TO 805
     DISTZ(1,1)=0.
     DO 801 IK=1,N
     DISTZ(IK+1,1)=DDO(IK)
 801 DISTZ(1,IK+1)=DDO(IK)
     DO 802 IM=1,N
     DO 802 IJ=1,N
 802 DISTZ(IM+1,IJ+1)=D(IM,IJ)
     XTM=1000.
     CALL IMPROT(LAB,N,XTM,DISTZ,DIST,DISS)
     DO 809 IH=1,N
 809 LAB1(IH)=L(LAB(IH))
     IFLAG=2
     IR=IR+1
     GO TO 44
805  IF(BTIME.GT.ITIM) GO TO 299
 901 IF(ILIST.NE.1) GO TO 902
 803 NUM=N/20
     ANUM=N/20.
     IF(ANUM.GT.NUM)NUM=NUM+1
     ICNT=0
     NON=0
 804 IR=IR+1
     ICNT=ICNT+1
     WRITE(6,808) IR,JC,(AREA(IR1),IR1=1,5),ICNT,NUM
 808 FORMAT('1',//,110X,'PAGE',I3,/,27X,'ROUTE NO.',I5,//,
    124X,5A4,//,23X,'ORDER OF CALL POINTS',//,13X,60('-'),//,
    213X,'BOX NO.',10X,'LOCATION',20X,'SHEET',I2,' OF',I2,//,
    313X,60('-'),//)
     NAN=NON+1
     NON=ICNT*20
     IF(NON.GT.N)NON=N
     DO 806 IW=NAN,NON
 806 WRITE(6,807)LAB1(IW),(SN(IW,IS),IS=1,10)
 807 FORMAT(15X,I5,5X,10A4,/)
     IF(ICNT.LT.NUM) GO TO 804
 902 IF(IPLOT.GT.1) GO TO 810
     CALL PLIK(DIST,NP1,LAB,X,Y,XORG,YORG,SCALE,JC,LAB1,
```

```
     1NPLOT)
  810 TDIST=TDIST+DIST
      ATDIST=TDIST*SCALE
      GO TO 997
  299 IR=IR+1
      WRITE(6,298) IR,JC
  298 FORMAT('1',110X,'PAGE',I3,15(/),15X,'** WARNING **',////,15X,
     1' TIME REQUIRED FOR ROUTE NO.',I3,
     2' GREATER THAN TIME ALLOWED',//,15X,
     3' TRY LESSER NUMBER OF BOXES IN ',
     4'THIS ROUTE',////)
      GO TO 901
  997 IF(JC.LT.M) GO TO 7
      WRITE (6,170) M,ATDIST
  170 FORMAT ('1',10X,'TOTAL NUMBER OF ROUTES',2X,I2,/11X,'TOTAL LENGTH
     1OF ROUTES',2X,F7.2,' MILES')
  999 STOP
      END
      SUBROUTINE IMPROT(NMROUT,MMR,TMM,DM,DIST,DISS)
      DIMENSION NMROUT(100),NEWRT(100),DM(101,101),NLR(100),LR(10,100)
C
C       THIS SUBROUTINE IMPROVES ROUTES BY REMOVING A CUSTOMER IN A ROUTE AND
C       REINSERTING IT IN THE BEST POSITION IN THIS ROUTE.
C
C
C      MMR      = # OF BOXES EN ROUTE
C      LM       = NUMBER OF THE ROUTE
C      DM(*,.)  = DISTANCE FROM (*) TO (.)
C      TMM      = DISTANCE REMAINING
C      I        = ROUTE INDEX
C      K        = BOX INDEX
C      ITSTCS   = TEST BOX NUMBER
C      NMROUT(.)= LABEL OF (.) IN CALL ORDER
C      DIST     = TRAVEL DISTANCE OF NEW ROUTE
C
C
C       STEP THRU ROUTES
C
      KM1=1
C
C       STEP THRU CUTOMERS OF ROUTE I
C
      DO 90 K=1,MMR
C
C       STORE ROUTE WITHOUT TEST CUSTOMER IN NEWRT(.)
C
      DO 10 KK1=1,MMR
      IF(KK1.LT.K) NEWRT(KK1)=NMROUT(KK1)
      IF(KK1.GT.K) NEWRT(KK1-1)=NMROUT(KK1)
   10 CONTINUE
```

```
      NEWRT(MMR)=0
      NLR(MMR)=0
C
C      TEST BOX IS K'TH BOX OF NMROUT(.)
C
      ITSTCS =NMROUT(K)+1
      KP1=NEWRT(K)+1
      IF(K.GT.1)KM1=NEWRT(K-1)+1
      IF(ITSTCS.EQ.KM1)GOTO 90
C
C        CALCULATE DISTANCE OF NEWRT
C
      TEMP=TMM-DM(KM1,ITSTCS)-DM(ITSTCS,KP1)+DM(KM1,KP1)
      ISVKK=0
      SVTM=TMM
C
C        KMP IS KM-PREVIOUS
C
      KMP=1
C
C        TEST CUSTOMER IN EACH POSITION AND SAVE LOCATION IN NEWRT (ISVKK) AND
C        DISTANCE ON NEWRT (SVTM).
C
      DO 20 KK2=1,MMR
      KMO=NEWRT(KK2)+1
      TEMPTM=TEMP+DM(KMP,ITSTCS)+DM(ITSTCS,KMO)-DM(KMP,KMO)
      KMP=KMO
      IF(TEMPTM.GE.SVTM) GOTO 20
      ISVKK=KK2
      SVTM=TEMPTM
 20   CONTINUE
C
C      IF NO CHANGE IN NEWRT TRY NEXT BOX
C
      IF(ISVKK.LE.1) GOTO 90
C
C        STORE NEW ROUTE
C
      TMM=SVTM
      ISVM=ISVKK-1
      DO 30 KK3=1,ISVM
      IF(KK3.LT.ISVKK) NMROUT(KK3)=NEWRT(KK3)
      IF(KK3.GT.ISVKK) NMROUT(KK3)=NEWRT(KK3-1)
 30   CONTINUE
      NMROUT(ISVKK)=ITSTCS-1
 90   CONTINUE
      DIST=0.0
      MNR=MMR-2
      DO 35 I=1,MNR
 35   DIST=DIST+DM(NMROUT(I)+1,NMROUT(I+1)+1)
```

```
      DISS=DIST+DM(1,NMROUT(1)+1)+DM(NMROUT(MMR),1)
      RETURN
      END
      SUBROUTINE PLIK(SOLT,NACCST,S,X,Y,XORG,YORG,
     1SCALE,JC,LAB1,NPLOT)
      DIMENSION JJJ(2),IDATE(3),X(100),Y(100),LAB1(100)
      INTEGER S(100)
      CALL TIME(5,0,IDATE)
C
C       PLOT AXIS
C
      CALL NUMBER(1.85,0.85,0.1,XORG,0.0,1)
      CALL PLOT(2.0,1.0,3)
      CALL PLOT(2.0,1.0,2)
      JJJ(1)=13
      YB=1.0
      YB1=YB-0.25
      DO 10 I=1,13
      AK=(XORG+(I*25))
      XB=(I*1.969)+2.0
      CALL SYMBOL(XB,YB,0.1,JJJ(1),0.0,-1)
 10   CALL NUMBER(XB,YB1,0.1,AK,0.0,1)
      CALL NUMBER(1.25,1.0,0.1,YORG,0.0,1)
      CALL PLOT(2.0,1.0,2)
      XB=2.0
      XB1=XB-0.75
      DO 20 I=1,10
      AY=(YORG+(I*25))
      YB=(I*1.969)+1.0
      CALL SYMBOL(XB,YB,.1,JJJ(1),90.0,-1)
 20   CALL NUMBER(XB1,YB,0.1,AY,0.0,1)
      CALL PLOT(2.0,1.0,2)
C
C      CHANGE COORDINATES TO MAP SCALE
C
      NN=NACCST-1
      DO 100 J=1,NN
      SX=X(J)
      X(J)=(SX-XORG)/2.54*2
      SY=Y(J)
 100  Y(J)=(SY-YORG)/2.54*2
C
C       PLOT ROUTES
C
      JJJ(2)=2
      XB=X(S(1))+2.0
      YB=Y(S(1))+1.0
      CALL PLOT(XB,YB,3)
      CALL SYMBOL(XB,YB,.1,JJJ(2),0.0,-2)
      X1=XB-0.25
```

```
      Y1=YB-0.20
      BLAB=LAB1(1)
      CALL NUMBER(X1,Y1,0.1,BLAB,0.0,-1)
      CALL PLOT(XB,YB,3)
      DO 50 J=2,NN
      IK=S(J)
      XB=X(IK)+2.0
      YB=Y(IK)+1.0
      X2=XB-0.25
      Y2=YB-0.20
      ALAB=LAB1(J)
      CALL SYMBOL(XB,YB,.1,JJJ(2),0.0,-2)
      CALL NUMBER(X2,Y2,0.1,ALAB,0.0,-1)
 50   CALL PLOT(XB,YB,3)
      IF(NPLOT.NE.1) GO TO 52
C
C      PLOT PRESENT ROUTE
C
      XB=X(1)+5.1
      YB=Y(1)+3.1
      CALL PLOT(XB,YB,3)
      CALL DASHLN(0.1,0.05,0.1,0.05)
      DO 51 J=2,NN
      XB=X(J)+5.1
      YB=Y(J)+3.1
 51   CALL PLOT(XB,YB,4)
C
C       PLOT HEADINGS
C
 52   SCALO=SCALE*2
      SCAL=SCALO*25.4
      CALL PLOT(15.0,5.0,3)
      CALL SYMBOL(15.0,1.25,.1,23HTOTAL TRAVEL DISTANCE  ,0.,23)
      CALL NUMBER(17.0,1.25,.1,SOLT,0.,1)
      CALL SYMBOL(15.0,1.50,.1,16HNUMBER OF BOXES ,0.,16)
      ANN=NN
      CALL NUMBER(17.0,1.50,.1,ANN,0.,-1)
      CALL SYMBOL(15.0,1.0,.1,17HSCALE :  1  MM = ,0.,17)
      CALL NUMBER(17.0,1.0,.1,SCALO,0.,2)
      CALL SYMBOL(18.0,1.0,.1,5HMILES,0.,5)
      CALL SYMBOL(15.5,0.75,.1,10H(1 INCH = ,0.,10)
      CALL NUMBER(17.5,0.75,.1,SCAL,0.,2)
      CALL SYMBOL(18.5,0.75,.1,6HMILES),0.,6)
      CALL PLOT(19.0,0.0,-3)
      RETURN
      END
      SUBROUTINE SELECT(L,X,Y,TIME,SNAME,NUM,TITLE,WAY,IGP)
      DIMENSION TITLE(20), WAY(20),LGP(100),L(100),X(100),Y(100),
     ASNAME(100,10),TIME(100),ID(10),D(10)
      READ(5,100) TITLE,WAY
```

```
100 FORMAT(20A4,/,20A4)
    READ(5,110) INO,IGP
110 FORMAT(2I4)
    READ(5,120) (LGP(K),K=1,IGP)
120 FORMAT(15I5)
    ATIME=.9
    J=1
    DO 190 II=1,NUM
    READ(7,140) IA,B,C,(D(K),K=1,10)
    IF(IA.EQ.LGP(J)) GO TO 200
    GO TO 190
200 L(J)=IA
    X(J)=B
    Y(J)=C
    TIME(J)=ATIME
    DO 201 IC=1,10
201 SNAME(J,IC)=D(IC)
    J=J+1
    IF(J.GT.IGP) GO TO 99
190 CONTINUE
140 FORMAT(2X,I3,2F10.2,3X,10A4)
 99 RETURN
    END
    SUBROUTINE COUNT(D,N,TITLE,WAY,SCALE)
    DIMENSION XLIST(5000),D(100,100),ICONT(11),INT(11),STOR(150)
    DIMENSION TITLE(20),WAY(20)
    REAL INT,INTER
    DATA STAR,BLNK/'*',' '/
    AMIN=99999.99
    AMAX=0.0
    NUMK=1
    SUM=0.
    ASUM=0.
    SCAL1=SCALE*5280
    DO 14 I=2,N
    K=I-1
    DO 14 J=1,K
    XLIST(NUMK)=D(I,J)
    IF(XLIST(NUMK).GT.AMAX) AMAX=XLIST(NUMK)
    IF(XLIST(NUMK).LT.AMIN) AMIN=XLIST(NUMK)
    SUM=SUM+XLIST(NUMK)
14  NUMK=NUMK+1
    NUMK=NUMK-1
    AVG=SUM/NUMK
    DO 16 I1=1,NUMK
    DSQ=(XLIST(I1)-AVG)**2
16  ASUM=ASUM+DSQ
    VAR=ASUM/(NUMK-1)
    STD=SQRT(VAR)
    TSUM=SUM
```

```
      SUM=TSUM*SCAL1
      TAVG=AVG
      AVG=TAVG*SCAL1
      TSTD=STD
      STD=TSTD*SCAL1
      BMAX=AMAX
      AMAX=BMAX*SCAL1
      BMIN=AMIN
      AMIN=BMIN*SCAL1
      RANGE=AMAX-AMIN
      INTER=RANGE/10
      DO 10 I2=1,10
10    ICONT(I2)=0
      INT(1)=AMIN*1
      DO 17 IG=2,11
17    INT(IG)=INT(IG-1)+INTER
      CALL SSORT(XLIST,NUMK,3)
      IKONT=0
      J1=2
      DO 19 JK=1,NUMK
      VALUE=XLIST(JK)*SCAL1
      IF(VALUE.GT.INT(J1)) GO TO 22
      IKONT=IKONT+1
      GO TO 19
22    ICONT(J1)=IKONT
      IKONT=1
      J1=J1+1
19    CONTINUE
      WRITE(6,100) TITLE,N,WAY
100   FORMAT('1',//,5X,20A4,//,5X,'STATISTICS OF DISTANCE MATRIX OF',
     A' GROUP',I4,' USING ',20A4,//)
      WRITE(6,101) SUM,AVG,STD
      WRITE(6,50) NUMK
50    FORMAT(20X,'TOTAL NUMBER OF ELEMENT IN DIST. MATRIX = ',I10)
      WRITE(6,51) AMIN,AMAX
51    FORMAT(20X,'MINIMUM DISTANCE = ',F10.2,'FEET',/,20X,'MAXIMUM',
     A' DISTANCE = ',F10.2,'FEET',//)
101   FORMAT(20X,'TOTAL DISTNCE = ',F16.2,'FEET',/,20X,'AVERAGE',
     A      ' DISTANCE = ',F14.2,'FEET',/,20X,'STANDARD ',
     B       'DEVIATION = ',F12.2,'FEET',///)
      WRITE(6,102)
102   FORMAT(30X,'**** FREQUENCY PLOT ****',///,20X,'INTERVALS',20X,
     A       'FREQUENCY',/)
      M=0
      DO 150 IJ=1,10
      IF(ICONT(IJ).EQ.0) GO TO 70
      M=ICONT(IJ)/4
      IF(M.EQ.0)M=1
      DO 9 JY=1,M
 9    STOR(JY)=STAR
```

```
 70   MN=M+1
      DO 8 JX=MN,150
8     STOR(JX)=BLNK
      WRITE(6,103) INT(IJ),ICONT(IJ),(STOR(K),K=1,60)
103   FORMAT(20X,F9.2,/,32X,I3,3X,60A1)
150   CONTINUE
      WRITE(6,152) INT(11)
152   FORMAT(20X,F9.2)
      WRITE(6,151) (NUMBE,NUMBE=40,240,40)
151   FORMAT(/,37X,6('I---------'),'I',/,37X,'0',6(7X,I3))
      RETURN
      END
$SIG
```

MULTIPLE ROUTE SCHEDULING BY SAVINGS ALGORITHM---
CLUSTER TRIAL RUN - NON-HIERARCHICAL - TDATA1

THE FOLLOWING ROUTING PRINT-OUTS FOR EVENLY DIST. DATA AREA ARE BASED ON : -

CAPACITY CONSTRAINT OF EACH TRUCK ROUTE	=	50 BOXES
TIME CONSTRAINT OF EACH ROUTE	=	90 MINUTES
AVERAGE SPEED OF TRUCKS EN ROUTE	=	15.00 M.P.H.
TOTAL NUMBER OF ROUTES IN THE AREA	=	2 ROUTES
TOTAL NUMBER OF BOXES IN THE AREA	=	80 BOXES
MAP SCALE	=	1 MM = 0.059652MILES
STOPPING TIME FOR RESIDENTIAL BOXES	=	45 SECONDS
STOPPING TIME FOR BUSINESS BOXES	=	AS SPECIFIED

CLUSTER INTERPRETATION - TDATA1 -GROUP 1

STATISTICS OF DISTANCE MATRIX OF GROUP 49 USING MEDIAN(GOWER)METHOD - CENTROID SORTING

TOTAL DISTNCE = 19090496.00FEET
AVERAGE DISTANCE = 16233.41FEET
STANDARD DEVIATION = 7973.69FEET

TOTAL NUMBER OF ELEMENT IN DIST. MATRIX = 1176
MINIMUM DISTANCE = 1259.85FEET
MAXIMUM DISTANCE = 41260.07FEET

**** FREQUENCY PLOT ****

INTERVALS	FREQUENCY	
1259.85	0	
5259.87	95	**********
9259.89	157	**********
13259.91	210	**********
17259.93	195	**********
21259.95	215	**********
25259.97	139	**********
29259.99	86	**********
33260.01	50	**********
37260.03	21	*****
41260.05		

I---------I---------I---------I---------I---------I---------I
0 40 80 120 160 200 240

MULTIPLE ROUTE SCHEDULING
USING TIME-SAVING METHOD

EVENLY DIST. DATA AREA
F.S.A. UNDEFINE
SLB ROUTES

** PRELIMINARY ROUTE **

ROUTE NO. 1

NO. OF BOXES EN ROUTE	=	49	BOXES
TOTAL STOPPING TIME	=	44.10	MINUTES
DISTANCE TRAVELLED(1ST TO LAST BOX)	=	38.65	MILES
TRAVEL TIME REQUIRED(1ST TO LAST BOX)	=	154.62	MINUTES
TOTAL TIME REQUIRED(1ST TO LAST BOX)	=	198.72	MINUTES
TOTAL DISTANCE TRAVELLED(STN TO STN)	=	39.61	MILES
TOTAL TRAVEL TIME(STN TO STN)	=	158.44	MINUTES
TOTAL TIME REQUIRED FOR THIS ROUTE	=	202.54	MINUTES

ORDER OF CALL POINTS:

STATION												
	22-	26-	19-	27-	29-	33-	38-	34-	28-	23-	24-	30-
	35-	43-	51-	49-	48-	42-	39-	41-	50-	45-	44-	40-
	36-	37-	32-	31-	25-	20-	16-	11-	10-	7-	1-	8-
	14-	18-	21-	17-	13-	6-	2-	5-	3-	4-	12-	9-
STATION												
	15-	0-										

MULTIPLE ROUTE SCHEDULING
USING TIME-SAVING METHOD

EVENLY DIST. DATA AREA
F.S.A. UNDEFINE
SLB ROUTES

** IMPROVED ROUTE **

ROUTE NO. 1

NO. OF BOXES EN ROUTE	=	49	BOXES
TOTAL STOPPING TIME	=	44.10	MINUTES
DISTANCE TRAVELLED(1ST TO LAST BOX)	=	36.33	MILES
TRAVEL TIME REQUIRED(1ST TO LAST BOX)	=	145.31	MINUTES
TOTAL TIME REQUIRED(1ST TO LAST BOX)	=	189.41	MINUTES
TOTAL DISTANCE TRAVELLED(STN TO STN)	=	38.95	MILES
TOTAL TRAVEL TIME(STN TO STN)	=	155.81	MINUTES
TOTAL TIME REQUIRED FOR THIS ROUTE	=	199.91	MINUTES

ORDER OF CALL POINTS:

STATION -												
	26-	19-	27-	29-	33-	34-	28-	22-	23-	24-	30-	35-
	38-	43-	51-	49-	48-	42-	39-	41-	50-	45-	44-	40-
	37-	36-	32-	31-	25-	20-	18-	11-	10-	7-	1-	8-
	14-	18-	21-	17-	13-	6-	2-	5-	3-	4-	12-	9-
STATION												
	15-	0-										