AN APPLICATION OF MULTIVARIATE ANALYSIS TO TIME OF DAY ROUTING IN TELECOMMUNICATION NETWORKS

by

ISABELLE SMITH

B.ENG. in Mechanical Engineering, McGill University, 1998

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Faculty of Commerce and Business Administration

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December 2000

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 or her 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 & Business Administration

The University of British Columbia
Vancouver, Canada

Date 15/12/00

DE-6 (2/88)

Abstract

The work presented in this thesis is a component of a larger project that is currently under way at the Center for Operations Excellence. This project was initiated in May of 1999. The main goal of this project is to help Telus reduce their yearly investments in the non-toll network by optimizing the current use of their telephone network. We chose to accomplish this by determining a new set of routing rules that would make the most efficient use of their current network. Three main approaches were selected to obtain this new set of routing rules. The first one was the development of a simulation that could serve as a test-bed to compare different sets of routing rules and to determine the optimal set. The second approach was the use of a linear program that would generate routing rules that minimize network utilization. Finally, we have approached the challenge of optimizing the network utilization by using multivariate analysis to help develop routing rules for time of day routing. This last approach is described in this thesis.

The objective of this thesis was to develop a methodology, using multivariate analysis, that would help us find if there were possible alternate routes that could be added to the current ones, which would take advantage of excess capacity in certain regions in the network.

This methodology was developed in SAS and included the use of Principal Components Analysis, Clustering Analysis and the development of an algorithm that would search for adjacent arcs that had sufficient available capacity to serve as alternate routes during certain periods of the day.

It was found that there are alternate routes that can be used during certain times of the day and that these routes can be found by using the methodology described in this thesis. This result is of great value to telephone companies as this means that there is a way for them to use existing capacity more efficiently and therefore avoid or delay investments into adding capacity to the telephone network.

Table of Contents

List of Tables

List of Figures

Acknowledgment

I would like to take the opportunity to acknowledge the help and efforts of the people who made it possible for me to complete this thesis.

First of all, I would like to thank my thesis supervisor, Professor Martin L. Puterman, for his advice and help in writing my thesis. His assistance and recommendations were greatly helpful. I would also like to thank Professor David Glenn for his encouragement, support and input as member of my thesis examining committee.

I extend sincere thanks to Jason Goto for his guidance and technical assistance. His help and suggestions were extremely valuable throughout this project.

I would like to thank the Centre for Operations Excellence (COE) for providing myself and other master students the opportunity to apply our skills and knowledge to industry challenges. In addition, the financial assistance provided by the COE was greatly appreciated. I would especially like to thank Stephen Jones for his assistance and support. Also, many thanks to all of the students and staff who were involved in this project.

I would finally like to express my sincere gratitude to my family and friends for their encouragement and support.

1 Introduction

The demand on a telephone network, i.e. the capacity that is used, fluctuates throughout the day. This demand varies depending on the area of the network as well as on the period of the day. For example, regions of the network that are located in business areas are usually very busy during the day and relatively calm during the evening, while the opposite is true for suburban areas. This means that at any given time, there is excess capacity in certain regions of the network. There is therefore a great potential for savings if this idle capacity could be used. In an effort to use this excess capacity, time of day routing rules can be put in place in order to distribute the demand more evenly across the network. This could be done, for example, by redirecting some of the demand away from the business areas and through the residential areas during the day.

1.1 Value of time of day routing

Taking advantage of time of day routing is of great value to a telephone company since this means that investments into adding capacity to the network can be avoided or delayed. Every year, telephone companies have to increase the capacity of their network in order to respond to the constantly growing demand on the network. This increase in the demand has been accelerated in the past years due to the increased Internet traffic, the special offers such as the long distance flat rate packages and the arrival of new services such as call fowarding. This increase in traffic forces telephone companies to add capacity to their network in order to sustain their level of service and remain competitive. Alternatively, telephone companies can find ways to use their currently available capacity more efficiently: this is the challenge that was presented to our group at the Center for Operations Excellence (COE).

1.2 Overview of the Telus project

The work presented in this thesis is a component of a larger project that is currently under way at the COE. This project was initiated in May of 1999. The main goal of this project is to help Telus reduce their yearly investments in the non-toll network by optimizing the current use of their telephone network. We chose to accomplish this by determining a new set of routing rules that would make the most efficient use of their current network. Three main approaches were selected to obtain this new set of routing rules. The first one was the development of a simulation that could serve as a test-bed to compare different sets of routing rules and to determine the optimal set. This simulation, which was programmed in C++, reproduces the network activity and can therefore be used to test different sets of routing rules. The inputs to the simulation are the call length and frequency distributions, the arc capacities and the routing rules. The main call statistics that are collected while the simulation is running are the number of blocked calls and the network utilisation. These call statistics can then be used as criteria to determine the optimal set of routing rules. The simulation can also permit us to determine problem areas in the network where it can be observed that the link capacity is not sufficient to handle all of the calls that are routed through that area.

The second approach was the use of a linear program that would generate routing rules that minimise network utilisation. A multi-commoditiy flow model that analyses a static interval of network activity was developed. This linear program generates a set of routing rules that minimises the network utilisation (i.e. it makes the most efficient use of the network currently in place given a specific data set). This approach has led to further research into using non-linear programming.

Finally, we have approached the challenge of optimizing the network utilization by using multivariate analysis to help develop routing rules for time of day routing. This is the approach that is described in this thesis.

1.3 Objective of this thesis

The objective of this thesis was to develop routing rules for time of day routing. More precisely, the objective was to develop a methodology, using multivariate analysis, that would help us find if there were possible alternate routes that could be added to the current ones, which would take advantage of excess capacity in certain regions in the network.

1.4 Scope of this thesis

The research for this thesis was performed on a 32-switch subset of the Telus circuit-switched non-toll network. Since Telus is moving toward using packet switching, the results of this thesis will only provide temporary benefits until packet switching is widely used. It is not possible to implement dynamic routing rules (i.e. routing rules which change dynamically based on the network utilization) in the subset of the network which was studied due to the fact that it contains more than one type of switch (i.e. it is not a homogeneous network). Therefore, static routing rules were developed. We have limited the number of routing rule tables to two. This means that during one part of the day, one of the routing tables will be used and during the other part, the other table will be used.

2 Background

2.1 Telecom Background

2.1.1 Terminology

The main terms that will be used in this thesis are explained below. Although the telecom industry is quite complex in nature, only a few terms and concepts need to be explained, as they are sufficient to understand this thesis.

Note: the numbers in brackets refer to Figure 1.

Switch (or node) (1)	A device that is used to connect circuits and make it possible for calls to be routed through the network
Trunk (or circuit) (2)	A communication link connecting two switches. It can carry only one call at a time.
Trunk group (or arc) (3)	Consists of a group of trunks between two switches, where each trunk can carry only one call at a time.
Arc capacity	Number of trunks that compose an arc. For example, the arc joining switch 01 to switch 02 has a capacity of 2.
AZ pair	Refers to a pair a switches that are connected directly by an arc. In the figure below, there are three AZ pairs: 01-02, 01-03 and 02-03
Arc utilization	Refers to the call volume that is carried on an arc (or trunk) during a certain period of time.
Centi-call seconds (or CCS)	Measures arc or trunk utilization (or call volume) in volumes of 100 seconds of call time. For example, if 1 trunk is used for 200 seconds during an hour, then the trunk utilization during that hour was of 2 CCS. Similarly, if 2 trunks on an arc were both used for 200 seconds during an hour, then the arc utilization on that arc was of 4 CCS.
Routing rules	Rules that determine the different paths through which a call can be routed from its origin to its destination (see routing rules section)
Blocked call	A call is blocked when all of the circuits on all of the possible routes are busy (i.e. are already in use)

Table 1 Terminology

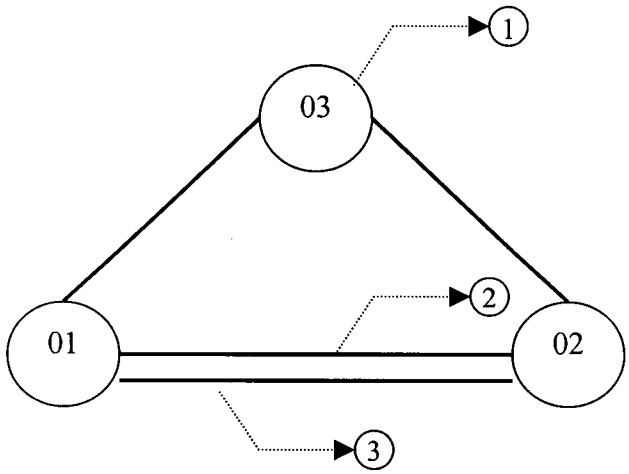


Figure 1 Terminology

2.1.2 Assumptions

There are three main assumptions that are made in the Telus project. These assumptions are described below.

- Each phone number is assigned to the end-office switch which serves that region and the telephone is connected to that switch through a smaller network composed of a series of devices and smaller switches. For the purpose of this thesis, we concentrate on the main telephone network composed of the end-office switches. Also, we assume that a call originates at the end-office switch where the call enters the main network. Similarly, we assume that it terminates at the end-office switch where the call leaves the network. To simplify the text, from now on the end-office switches will be referred to simply as switches, as the work in this thesis is focused solely on the non-toll network composed of end-office switches.
- In addition to the main network, there is an overlaying network which is responsible for directing the long distance calls. Similarly, we assume that a long distance call originates at the switch where if first enters the non-toll network and that it terminates

where it exits the non-toll network. Therefore, a long distance call is treated as a local call based on the portion of the non-toll network that is uses.

- Switches also have a limited capacity: they cannot handle more than a certain number of calls at any given time. However, since capacity constraints usually occur due to trunk capacity limits, we assume that the switch capacities are unlimited.

2.1.3 Routing rules

A brief description of routing rules is given in this section. A more thorough description of these routing rules can be found in the master thesis written by Darrel Braun[1], a former member of the COE Telus project.

Routing rules are rules that dictate which paths a call can use when it is routed from a specific origin to a specific destination across the network. All the routes that are possible paths between an origin and a destination switch are summarised into a routing table. The network that is being studied for this project is composed of 32 switches that are almost all interconnected by arcs. These switches are divided into a West and East sector, depending on their location. Two of these switches, one in the East sector and one in the West sector, can serve as tandems. As it is shown in Figure 2, knowing the sector in which a switch is located is important because this determines how the calls will be routed. The routing rules, which are used in this 32-node network, are based on **fixed hierarchical routing**.

When a call is placed, it is routed to the destination switch through fixed hierarchical routing rules. These rules vary depending on whether the origin and the destination switches are located in the same sector (see Figure 2).

[1] Darrel Braun, *Efficient routing of telephone calls in a circuit-switched network*, 2000.

Routing between switches within a sector

If the two switches are in the same sector, then a call would be routed using the following rules. The call would first be routed from the origin switch to the destination switch using the Primary High Usage (PHU) route, i.e. the trunk group connecting the origin and the destination switches directly. If all of the circuits in this trunk group were busy, i.e. they were all carrying calls, then the call would be routed through the Intermediate High Usage (IHU). The IHU is the name given to the route joining the origin switch to the tandem switch in its sector and then to the destination switch. If all of the trunks on this last route were busy, then the call would be blocked (the person making the call would hear a fast-busy signal and would have to try calling later).

Routing between switches in different sectors

These routing rules are slightly different in the case where a call is routed between an origin switch and a destination switch located in different sectors. In this case, the call would once again first be routed through the PHU. However, if this first route were busy, the call would be routed through the tandem in the sector opposite to that of the origin switch. This second route is called the IHU. The last route through which the call could be routed is called the Alternate Final (AF). This rule routes the call from the origin, through the two tandems and finally to the destination switch. All of these rules are summarized in Figure 2.

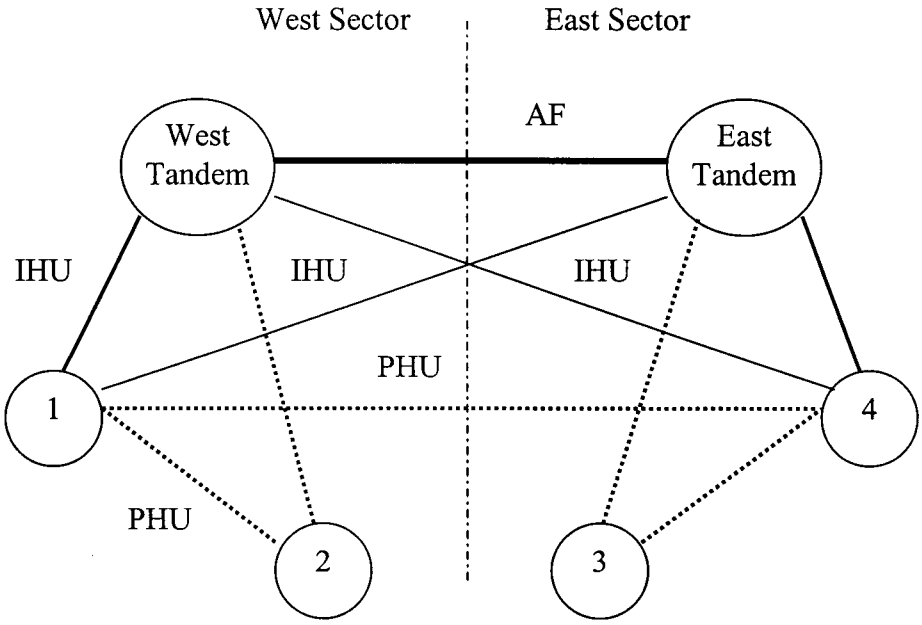


Figure 2 Fixed Hierarchical Routing rules

The routing rules between an origin and a destination switch are then summarized into a routing table. In this table, the possible routes are listed in the order in which they should be tried. For example, if we look at Figure 2, the possible routing rules to route calls from switch 1 to 4 would be summarized in the routing table as follows:

Origin	Tandem 1	Tandem 2	Destination	
01	-	-	04	← First route
01	East Tandem	-	04	← Second route
01	West Tandem	East Tandem	04	← Third route

Table 2 Example of a routing table

The routing table shown above, in Table 2, indicates that when calls are routed from the origin switch to the destination switch, the call would first be routed on the direct arc connecting the origin to the destination (the arc 01-04). If all of the lines on this first route were busy, then the call would be routed using the second route. This second route would direct the call from the origin switch to the east tandem and then from the east

tandem to the destination switch (i.e. the route would be 01 - "east tandem" - 04). Finally, if this second route were busy as well, then the call would be routed using the third route. This third route would direct the call from the origin to the west tandem, then from the west tandem to the east tandem and finally from the east tandem to the destination (i.e. the route would be 01-"west tandem"- "east tandem" – 04).

An alternative to these fixed hierarchical routing rules would be to use **dynamic routing rules**. This would mean that the possible routes would change depending on the time of the day. These routes take into account the network activity throughout the day and are used to direct the call traffic away from busy areas. These routing rules make more efficient use of the network, i.e. the amount of idle capacity is minimized. However, dynamic of routing rules cannot be implemented on the TELUS network as it is not a homogeneous network (i.e. it has more than one type of switches). Dynamic routing requires that the network be composed of only one type of switches because it uses a switch function that is not compatible from one switch type to another.

Another alternative, and this is what we are investigating in this thesis, is to have more than one routing table. For example, one set of routing rules (or routing table) could be used during the morning and another one could be used for the rest of the day. These routes would still follow fixed hierarchical rules, but they would be adapted based on the usual network activity during that period of the day. These new routes may require that some switches, other than the two main tandem switches, be used as tandems. This is

possible as every switch has the capacity to tandem calls. The objective of this thesis was to develop these time of day routing rules.

2.1.4 Demand Types

The demand on a network, i.e. the number of calls carried at a certain point in time, fluctuates continuously. The demand across the network varies with the time of day, day of week, and with holidays. For example, Mothers' day has the highest demand of the year. The demand also varies based on the origin and the destination switch. It also varies due to holidays. An example of how this demand varies is shown in Figure 3. This figure represents the variation of the demand, at an aggregate level for a subset of the network, for 16 days. In this example, we can see that the weekdays are generally busier than the weekend (except for the first Tuesday, which is Remembrance day).

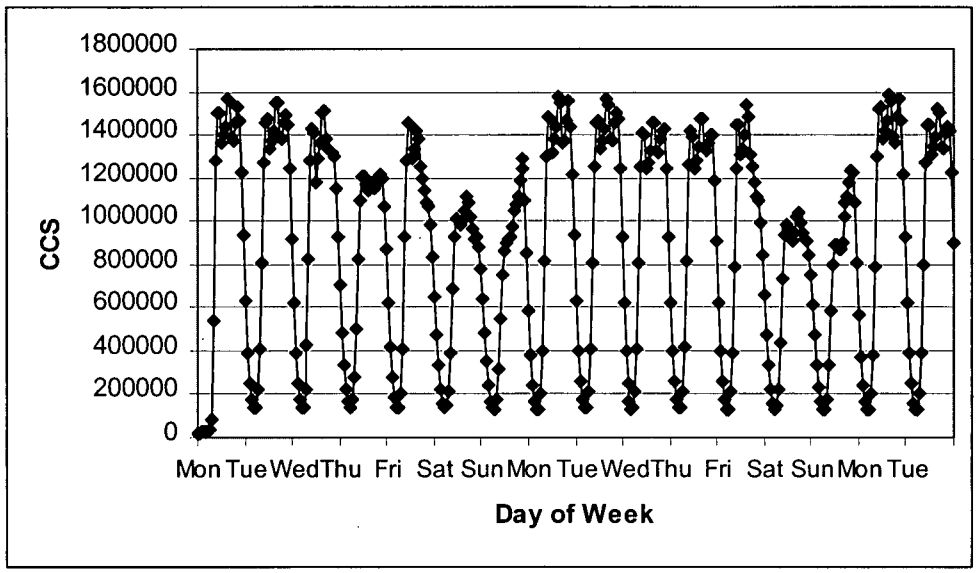


Figure 3 Variation of the demand during the week

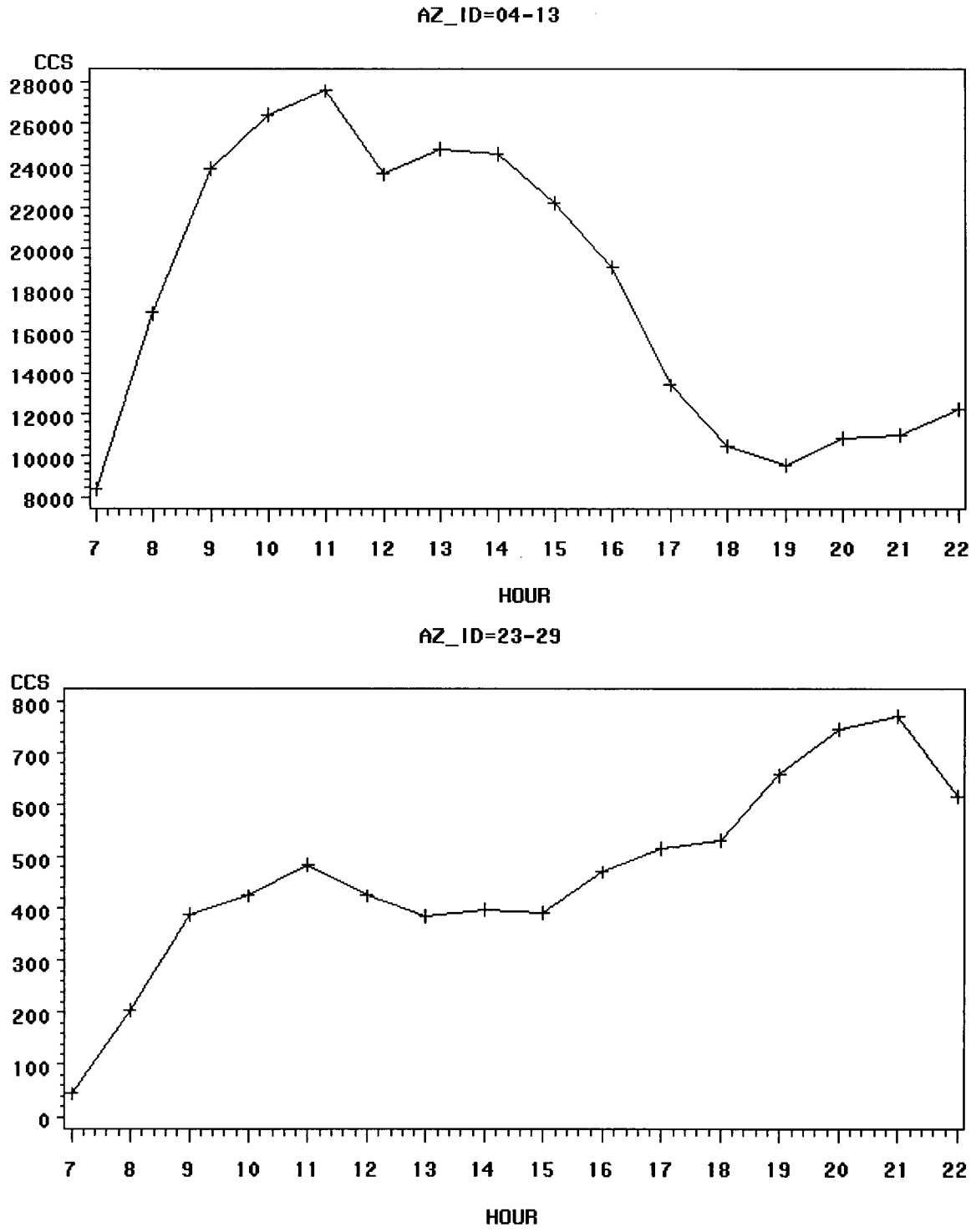


Figure 4 Variation of arc utilization with time of day

In the top graph in Figure 4, the arc utilization is very high in the morning, relatively high in the afternoon and it is low in the evening. These two switches are located in business areas. The opposite pattern is shown in the bottom graph of Figure 4, where the arc utilization is low in the morning and in the afternoon, but it is high in the evening. The switches 23 an 29 are located in residential areas. On each of these arcs, we can see that there is excess capacity during some periods of the day. As it was mentioned earlier, the object of this thesis was to find routing rules that would take advantage of these differences in arc utilization.

2.2 Technical Background

2.2.1 Multivariate Methods

Some multivariate methods are used to identify similarities among observations in a data set with many variables. For example, in the case of this thesis, multivariate methods are used to find similarities among the demand patterns of the arcs in the network. Multivariate methods are especially useful in the cases where the data sets consist of very large numbers of observations and contain many variables. This is the case for the data analyzed in this thesis. There were 460 observations representing pairs of switches on which the arc utilization had been recorded for 24 consecutive hours. Two multivariate methods are used in this thesis: principal components analysis and cluster analysis. These two methods are defined in the following sections.

2.2.2 Principal Components Analysis

Objectives of the PCA

The main objective of the PCA is to identify new variables that will help determine the true dimensionality of the data set while keeping most of the information. These new

variables can then be used for further analysis such as the cluster analysis used in this thesis.

When a PCA is performed, the number of principal components created is equal to the number of original variables. However, these new variables are now uncorrelated. In addition, the first principal component is the most important since it explains most of the variability. Each subsequent principal component is less important as it explain less and less of the variability. The number of variables is reduced by eliminating the principal components that have little variability.

Definitions behind the PCA

Briefly, performing a PCA "involves a mathematical procedure that transforms a set of correlated variables into a smaller set of uncorrelated variables called principal components"[2]. These principal components are obtained as follows.

In the following definitions, x is a vector chosen randomly from a population, where

$$x = [x_1, x_2, ..., x_p]^T$$

and the vector population mean is designated by μ.

The **first principal component** is defined by $y_1 = a_1^T(x - \mu)$. This equation can be rewritten in the following form:

$$y_1 = a_{11}(x_1 - \mu) + a_{12}(x_2 - \mu) + ... + a_{1p}(x_p - \mu), \text{ where}$$

the vector a_1 is chosen such that the variance of $a_1^T(x - \mu)$ is the greatest and a_1 is subject to the constraint: $a_1^T a_1 = a_{11}^2 + a_{12}^2 + ... + a_{1p}^2 = 1$.

[2] Johnson, p.3.

The variance of the first principal component is equal to the largest eigenvalue, λ_1, of the variance-covariance matrix, Σ. This relation can be written as:

$$\text{Var}(y_1) = \text{Var}(a_1^T(x-\mu)) = \lambda_1.$$

The **second principal component** is defined similarly by $y_2 = a_2^T(x-\mu)$, where a_2 is chosen such that the variance of $a_2^T(x-\mu)$ is the greatest except that now, the vector a_2 is subject to the two following constraints: $a_2^T a_2 = a_{21}^2 + a_{22}^2 + \ldots + a_{1p}^2 = 1$ and $a_2^T a_1 = 0$

The second constraint means that the second principal component is uncorrelated with the first.

The variance of the second principal component is equal to the second largest eigenvalue, λ_2, of the variance-covariance matrix, Σ. This relation can be written as:

$$\text{Var}(y_2) = \text{Var}(a_2^T(x-\mu)) = \lambda_2.$$

The subsequent principal components are defined in a similar manner. This definition can be generalized as follows:

The **jth principal component** (j=3, 4, …, p) is defined by $y_j = a_j^T(x-\mu)$, where a_j is chosen such that the variance of $a_j^T(x-\mu)$ is the greatest and the vector a_j is subject to the two following constraints:

$a_j^T a_j = a_{j1}^2 + a_{j2}^2 + \ldots + a_{jp}^2 = 1$ and $a_j^T a_k = 0$, where k=1, 2 , …, j-1.

The variance of the jth principal component is equal to the jth largest eigenvalue, λ_j, of the variance-covariance matrix, Σ. This relation can be written as:

$$\text{Var}(y_j) = \text{Var}(a_j^T(x-\mu)) = \lambda_j.$$

Therefore, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$ represent the eigenvalues of Σ and $a_1, a_2, \ldots, a_p$, represent the normalized eigenvectors of Σ. As the eigenvectors, a_j, are normalized to have a length of 1, the sum of the squares of all of the coefficients or weights composing this vector is equal to 1. In order to determine how much of the variability is captured by each of the principal component variable, we need to evaluate the ratio $\lambda_j/tr(\Sigma)$, where

$j = 1, 2, \ldots, p$

λ_j = variance accounted for by the j^{th} principal component

$tr(\Sigma)=\lambda_1 + \lambda_2 + \ldots + \lambda_p$

Therefore, the ratio $\lambda_j/tr(\Sigma)$ represents the ratio of the variance accounted for by the j^{th} component with respect to the total variability.

Principal Components Scores

The principal component scores are defined as the values of the principal component variable for each observation and are calculated as follows:

$$y_{rj} = a_{jr}(x_r - \mu), \text{ for } j = 1, 2, \ldots, p \text{ and } r = 1, 2, \ldots, N$$

where,

y_{rj} = the j^{th} principal component score for the r^{th} observation

The principal components can either be calculated from the correlation matrix or the variance-covariance matrix of the data. However, these can give extremely different results. The variance-covariance matrix should not be used unless the units for each variable are comparable and the variables have been standardized in some way. If the

correlation matrix is used, the equations described above remain the same, except the mean of the variables, μ, is not included in the calculations.

How many principal components should be retained?

As we wanted to reduce the number of variables in our data set, we needed to determine how many principal components should be kept. The first few principal components usually capture most of the variability while the rest of them do not bring any more valuable information about the data. Therefore, we must keep only meaningful principal components as we are trying to reduce the dimensionality of the data. There are three main criteria that can be used to decide how many principal components to keep: the eigenvalue-one criterion, the SCREE test and the proportion of variance that is accounted for by each principal component. The eigenvalue-one criterion should only be used if the computations to obtain the principal components were carried out on the correlation matrix. The two other criteria can be used in either case.

A. Eigenvalue-one criterion

Using this criterion, only the principal components with eigenvalues greater than 1.00 are kept. The reason for doing this is that components with eigenvalues less than 1.00 do not account for much of the variance and are therefore considered as trivial and are dropped. This criterion should never be used when performing the PCA on the raw data or the variance-covariance matrix.

B. SCREE test

Creating a SCREE plot can help in making this decision. This plot is composed of the eigenvalues (on the y-axis) versus the eigenvalue number (on the x-axis). This is illustrated in Figure 5. From this plot, we can see that the first three points have large

eigenvalues while the rest of them are small and tend to level off. These last points can be dropped as their eigenvalues are small and therefore, they do not explain much of the variability. When using a SCREE plot, it is assumed that the true dimensionality of the data is equal to the number of points that are not dropped. In this example, the SCREE plot would therefore suggest that the true dimensionality of the data is equal to three.

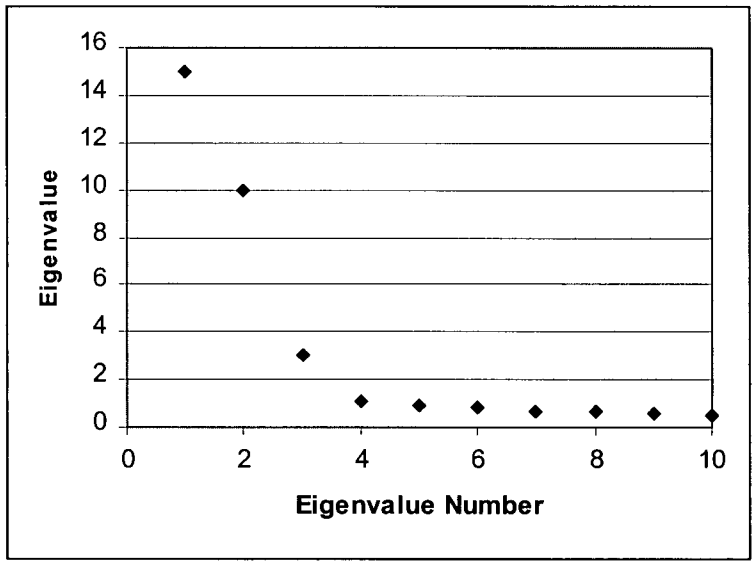


Figure 5 SCREE plot – Example

C. Proportion of the variance accounted for

As it has been explained earlier, the ratio $\lambda_j/tr(\Sigma)$ (i.e. the variance accounted for by a specific principal component divided by total of the eigenvalues of the variance-covariance matrix) can be used to determine how much of the variance is captured by each of the principal components. This ratio can be used as a selection criterion. For example, a researcher could decide to retain only the principal components which account for more than 5% of the total variance.

Using PCA for screening data

It is possible to use Principal Components Analysis (PCA) to screen the data. PCA can help identify if there are any abnormalities, special patterns or outliers within the data (i.e. observations whose measured variables may seem inconsistent compared to those on other observations). These peculiarities about the data may not be obvious when looking at the measured variables. However, once a PCA is performed, it may become quite evident. In brief, while performing a PCA, new variables called principal component scores are created. In turn, the principal component scores can be used to create graphs that will enable the researcher to determine if there are any abnormalities with the data, as these will affect further analyses. In the case where outliers are found, they should be removed prior to pursuing the analysis.

Using the results of PCA for clustering

In addition to being very useful for screening data, PCA is very helpful when performing clustering (i.e. dividing the observations from a data set into subgroups that show similar characteristics). Instead of using all of the variables provided in the data set, the dimensionality of the data set can be reduced by using PCA and the new principal component scores obtained from the PCA can then be used as inputs to the cluster analysis.

2.2.3 Cluster analysis

Objectives of Cluster Analysis

Cluster analysis is a method used to make sense out of large and complicated data sets, which contain a large number of observations with many measured variables. Cluster analysis is used to separate observations into subgroups based on the similarities of their

measured variables. If meaningful clusters are formed, then it becomes possible to analyze the behavior displayed by each group. This method is therefore very useful since it reduces the analysis effort by decreasing the number of observations to a few groups. Clustering methods can be divided into two main categories depending on whether they are hierarchical or nonhierarchical in nature.

Non-hierarchical Clustering Methods

These non-hierarchical clustering methods are initiated by first selecting a set of points that will serve as cluster seed points and then assigning each observation point to its closest seed point. The distances between each observation point and the seed points are calculated by using one of the dissimilarities or distance measure method that is described further in this thesis. Once this first step is finished, the size of each cluster and the distances between each pair of clusters are evaluated. Based on specified criteria, when a cluster is too large it is split and when some clusters are too close to each other, they are joined.

The FASTCLUS procedure in SAS (see Appendices for more details) is an example of a non-hierarchical clustering method. The way this procedure works can be summarized as follows:

- A set of cluster seed points are selected as initial approximations of the means of the clusters.
- Each point in the data set is temporarily assigned to its closest seed.
- The seed points are replaced by the mean of the temporary clusters.
- These two last steps are repeated until the clusters do no change anymore.

Although these are good sorting methods, they are very dependent on the initial set of seed points and on the specified number of clusters. Therefore, many different solutions could be obtained from the same data set.

Hierarchical Clustering Methods

When a hierarchical clustering method is used, each observation initially belongs to a cluster by itself (i.e. if there are 40 observations, there will be 40 clusters). Then, the two closest clusters are joined to form a new cluster (the two original clusters are therefore removed). This step is repeated until all of the clusters are merged into one single cluster. The hierarchical methods differ in the way that the distance between the clusters is calculated (see next section, Distance or Dissimilarity Measures). The SAS CLUSTER procedure makes it possible to chose among different hierarchical clustering methods. The more common methods are summarized Table 3[3]:

Method	Distance between two clusters is defined as...
Average Linkage	The average distance between all the possible pairs of observations, one in each cluster
Centroid Method	Euclidean distance (or squared Euclidean distance) between the centroids or the means of the two clusters
Complete Linkage	Distance between their furthest members
Single Linkage (nearest neighbour method)	Distance between their closest members
Ward's Minimum Variance Method	(Square of the distance between the cluster means)/(sum of the reciprocals of the number of points within each cluster)

Table 3 Common methods used for hierarchical clustering

[3] SAS/STAT, volume 1, pp.530 to 636 and Johnson pp.322 to 327.

Distance or Dissimilarities Measures

The main measures that are used to determine the similarities or distances between the points are the Euclidean distance and the Squared Euclidean distance. The Euclidean distance is simply the length of the straight line which joins two points. This is calculated as follows:

- The distance between a point P1 (x_1, y_1) and a point P2 (x_2, y_2) is equal to $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- The distance between two points in a multidimensional space is equal to $\sqrt{\sum_{i=1}^{N}(p_{1i} - p_{2i})^2}$, for i=1 to the number of dimensions and p_{1i} represents the coordinate of point 1 in dimension i and p_{2i} represents the coordinate of point 2 in dimension i.

How many clusters should be retained?

The hierarchical clustering methods require that we decide how many clusters to keep. The total number of clusters created by hierarchical clustering methods is equal to the number of observations minus one. This number of clusters must be reduced to a number of clusters that is neither too small (this would oversimplify the data) or too large (this would separate observations that are relatively similar into several groups, thus creating more groups than necessary). In order to determine the appropriate number to retain, the Cubic Clustering Criterion (CCC), the Pseudo Hotelling's T^2 Test (PST2) and the Pseudo F statistics (PSF) results can be examined.

The values of PST2 are a good indication of whether or not the combination of the two clusters should have been done. When the value of PST2 for a particular combination is

small relatively to the other values, this indicates that the clusters should be combined. If it is large, they should probably not. The PST2 is qualified as small when it is small relative to the others.

The CCC can help determine the number of clusters to retain when it is plotted against the number of clusters. On this plot, the peaks that have values of CCC that are greater than 3 indicate that they correspond to suitable number of clusters.

Another criterion that can be used to judge the number of clusters in a data set is to look at the pseudo *F* statistic (PSF). Relatively large values indicate that the number of clusters is probably appropriate.

Another way of determining the optimal number of clusters is to look at the tree diagram generated from the cluster analysis results. From this diagram, it is sometimes possible to determine visually into how many clusters the observations seem to fall. An example of a tree diagram is shown in

Figure 6. This diagrams shows how the clusters were formed. The first level of the diagram shows as many clusters as there are observations. As we look higher into the graph, we can see how the different clusters were merged sequentially until all of the clusters are merged into one. This tree diagram shows that there are three main clusters. These are circled on the diagram.

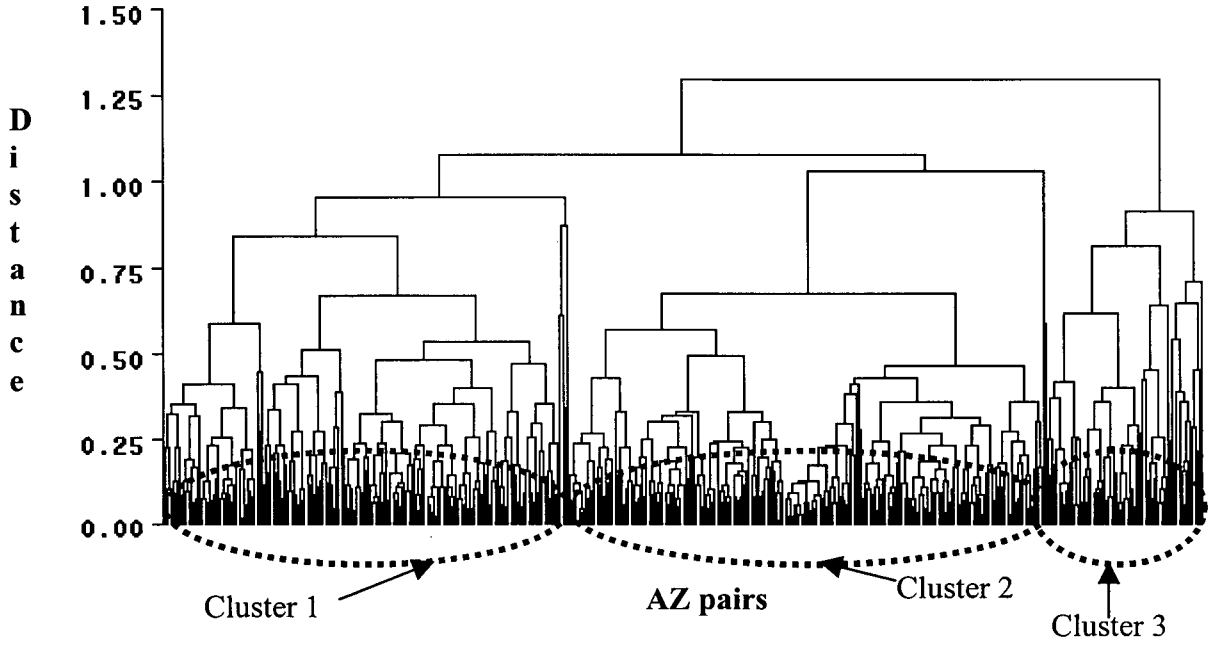


Figure 6 Example of a tree diagram

3 Methodology

3.1 Overview of the approach

The following example, illustrated in Figure 7, gives an overview of the approach used in this thesis. Suppose that we have an arc between switches A and B which is highly utilized during the afternoon, but which is less busy during the morning. We are looking for two adjacent arcs that display a complementary demand pattern, i.e. arcs which are less utilized during the afternoon.

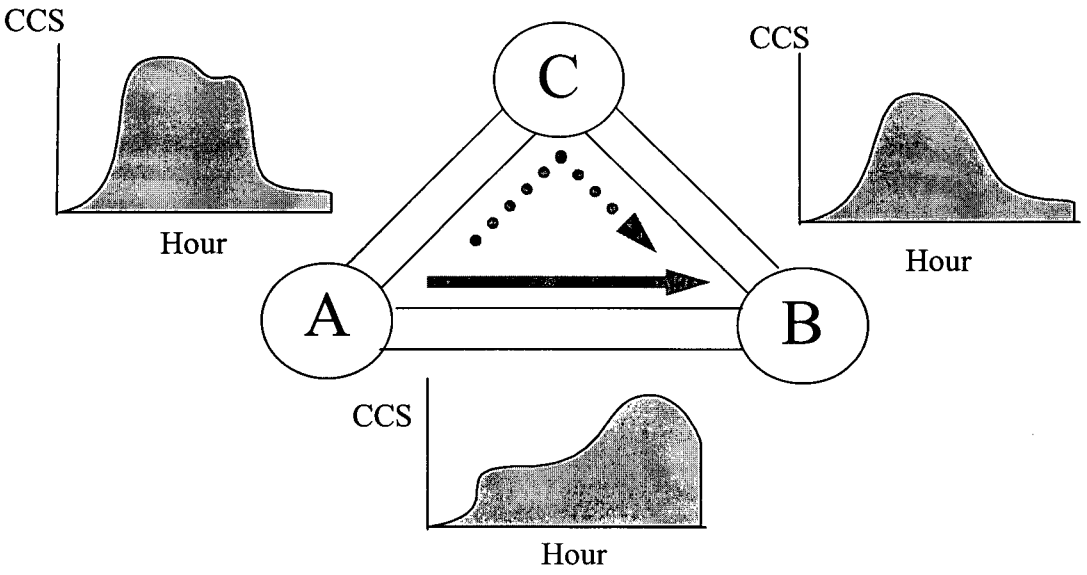


Figure 7 Example

Once these arcs have been identified, we can then investigate to see if there is sufficient capacity on the pair of adjacent arcs to justify adding them as a new alternate route. If there is enough capacity, then this new alternate route can be added to the routing table during the period of the day when it is needed. For example, suppose that we find that arcs 01-07 and 04-07 could be used as an alternate route during the morning between the origin switch 01 and the destination switch 04. In other words, suppose that these arcs have enough available capacity to carry the traffic that would normally be blocked

between 01 and 04, then this new route can be added to the routing table as a last alternate route (see Figure 8). This new alternate route would only be used in the event that all the circuits are busy on the first three routes.

Origin	Tandem 1	Tandem 2	Destination	
01	-	-	04	← First route
01	East Tandem	-	04	← Second route
01	West Tandem	East Tandem	04	← Third route
01	07	-	04	← New alternate route

Figure 8 Routing table with new alternate route

The methodology used to obtain these new alternate routes can be divided into four main steps. The first one is to prepare the data set so that it is meaningful and it can be analyzed. The second one is to perform a PCA. This step is performed to reduce the dimensionality of the problem. Once we have decided how many principal components to keep, these new variables are used as inputs to the next step: the cluster analysis. The cluster analysis is performed to obtain groups which display similar demand patterns (e.g. group together AZ pairs which have a busy morning compared to a low afternoon and low evening). This step is necessary to find pairs of arcs that could be used as new alternate routes. These arcs have to be adjacent and have a demand pattern that is complementary to that of the Primary High Usage arc. The last step is to determine if there is enough capacity on these new routes to carry the extra traffic. All these steps were implemented by code programmed in SAS (see Appendix L).

3.2 Overview of the methodology

The methodology used in this thesis was developed in SAS (Statistical Analysis Software). An overview of the numerous steps involved in finding new alternate routes is shown in Figure 9 and Figure 10. These steps are described more thoroughly in the next sections.

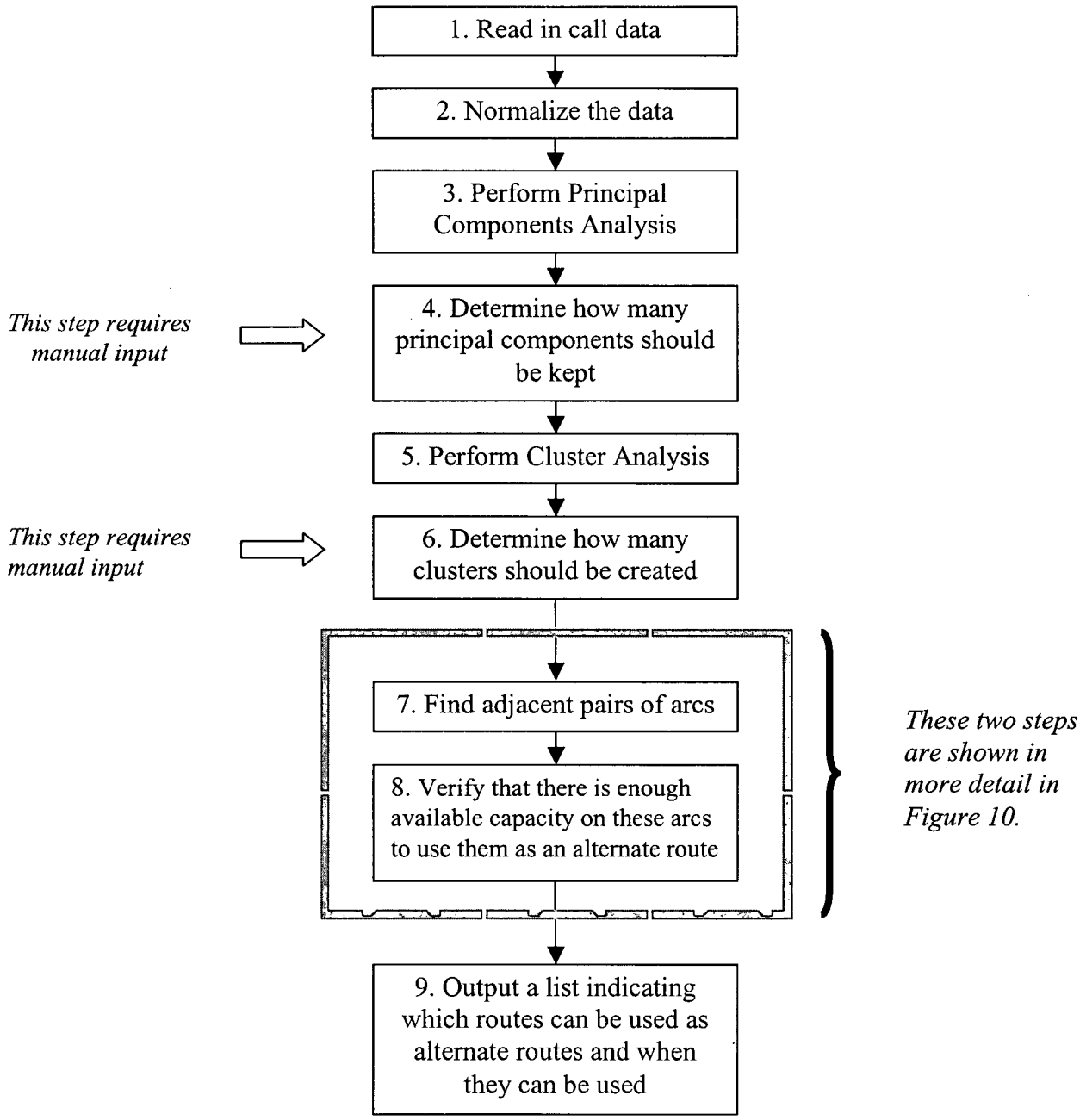


Figure 9 Overview of the methodology

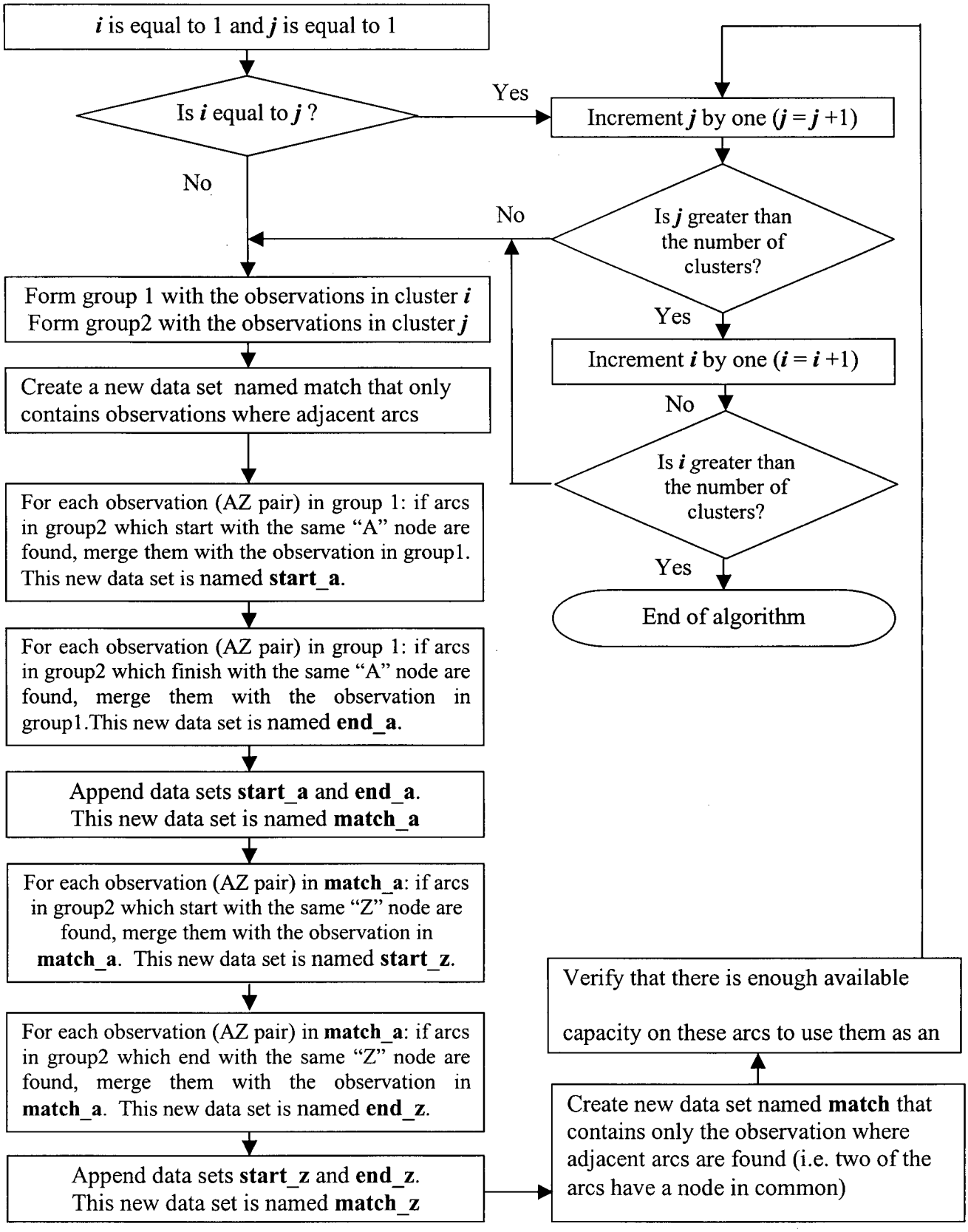


Figure 10 Overview of algorithm used to find new routes

3.3 Data preparation

The data used for this analysis were a subset of the data that were provided to us by Telus. The data included summarized call volumes (in call seconds) on an hourly basis for each AZ pair. This analysis was performed on data obtained for a particular day, July 31st 2000, but it could easily be repeated for data of any other day. In order to protect the confidentiality of the data, the name of the 32 switches were replaced by numbers ranging from 0 to 31. The data set was then transposed as shown in Table 4. This table represents the measured variables for one of the 460 observations (AZ pairs). For each AZ pair, the total call volumes in CCS, i.e. the total traffic for both directions, was added and summarized on an hourly basis. For example, the value under H8, i.e. 2498.4, for the AZ pair 0-1 represents the sum of all the traffic carried from 0 to 1 and from 1 to 0 during the 8:00AM to 9:00AM period.

Obs	AZ_ID	A	Z	_NAME	H0	H1	H2	H3	H4	H5
1	00-01	0	1	CCS	110.8	97.9	92.8	114.0	172.5	240.5

Obs	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15
1	384.4	966.7	2498.4	4217.3	4419.4	4568.2	3356.7	3695.8	3928.0	3696.8

Obs	H16	H17	H18	H19	H20	H21	H22	H23
1	2443.2	962.6	551.7	406.5	373.3	372.7	243.4	199.9

Table 4 Sample of data used in this analysis

The data contained in Table 4 were then normalized for each AZ pair. The hourly call volumes were scaled for each AZ pair so that the values of the call volumes would range from 0 (for the minimum hourly call volume) to 1 (for the maximum hourly call volume). This was done so that it would be possible to compare the demand patterns of each AZ pairs even though they did not have the same level of utilization. We are interested in the shape of the demand patterns, not in the volume itself. In addition, only the data for the time period between 7:00AM and 22:00PM were kept as the demand on the network is

generally low for the rest of the time and there is therefore no need to look for alternate route.

3.4 Principal Components Analysis

The first step that we took in our analysis of the data was to screen the data using Principal Components analysis. This analysis was performed on the normalized data using the PRINCOMP procedure in SAS. The reason why this analysis was performed on the normalized data is that our goal is to find similar demand patterns in the data, rather than finding similar call volume levels. The difference in arc utilization from one arc to another during the same hour can be as large as 27530.8 CCS. Therefore, performing a PCA on the non-normalized data might only try to differentiate arcs based on the arc utilization (i.e. arcs that generally have a high arc utilization would be grouped together while arcs with smaller capacities and therefore lower arc utilization would be grouped together). A PCA was performed on the non-normalized data to show that this was the case (see Appendix B).

Once the PCA was done, we decided how many principal components needed to be kept based on the eigenvalue-one criterion, the SCREE test and the proportion of variance that is accounted for by each principal component (see Background section for more details about these criteria). The results obtained from the PCA and the evaluation of the criteria are shown in the Results section.

3.5 Cluster analysis

The cluster analysis was then performed using the FASTCLUS and the CLUSTER procedures in SAS (see Appendix A for more details about these procedures). These

analyses used as inputs the variables (the principal components) that were obtained from the PCA.

Clustering using the FASTCLUS procedure:

Using the FASTCLUS procedure was a difficult task as it requires that the user defines how many clusters to form and this was not obvious from looking at the data. The cluster analysis using the FASTCLUS was repeated several times, specifying different number of clusters, but it was difficult to conclude from this analysis what was the ideal number of clusters that should be used.

Clustering using the CLUSTER procedure

The clustering was performed again using the CLUSTER procedure in SAS. In order to determine the appropriate number of clusters, the Cubic Clustering Criterion (CCC), the Pseudo Hotelling's T^2 Test (PST2) and the Pseudo F statistics (PSF) results were examined (these criteria are described in the Background section).

3.6 Search for pair of adjacent arcs

After the AZ pairs were grouped using the cluster analysis, those groups were used to search for pairs of adjacent arcs. This search is illustrated as follows. Suppose that we are analyzing the two groups shown in Table 5. We first proceed by finding, for each AZ pair in group1, arcs in group 2 which either start or end with the same node as the "A" node of the AZ pair. Then we search, once again for each AZ pair in group 1, for arcs in group 2 which either start or end by the same node as the "Z" node of the AZ pair. By doing this, a table similar to that in Table 6 is obtained.

Group1:

Observation	A	Z	AZ_id
1	01	07	01-07
2	02	08	02-08
3	04	11	04-11

Group2:

Observation	A	Z	AZ_id
1	01	11	01-11
2	01	17	01-17
3	04	05	04-05
4	05	11	05-11

Table 5 Example of search for pairs of adjacent arcs.

Observation	A	Z	AZ_id	Starts or ends with A	Starts or ends with Z
1	01	07	01-07	01-11	.
2	01	07	01-07	01-17	.
3	02	08	02-08	.	.
4	04	11	04-11	04-05	05-11

Table 6 Adjacent arcs

The next step is to determine if any of the arcs which start or end with the "A" node of the AZ pair have a node in common with any of the arcs which start or end with the "Z" node of the AZ pair. In the example above, there is a match only for observation 4, i.e. the AZ pair 04-11. This means that the arcs 04-05 and 05-11, belonging to group 2, are a pair of adjacent arcs that could possibly be used as an alternate route for the AZ pair 04-11. However, before adding these new routes to the routing tables, we first need to

determine if and when they have enough available capacity in order to carry the extra traffic offered by the AZ pair.

3.7 Feasibility test

Once the file containing all of the adjacent pairs of arcs was created, we then verified whether or not there was sufficient available capacity on the new alternate route to carry the extra traffic. This was done by calculating the minimum available capacity. Since there were no accurate data available concerning the capacity of each arc at the time when the code for this thesis was developed, the capacity of each arc was assumed equal to the maximum arc utilization during the day that was studied. It was then assumed that an arc could only be used as part of an alternate route if there was more than 30% of its capacity available. This seemed like a reasonable assumption as we want to make sure that we will not be incurring any blockage on the arcs forming the new routes: this would change the network flow and may create other problem areas. The hourly available capacity was therefore calculated by subtracting the hourly arc utilization from 70% of the maximum daily arc utilization (see Figure 11).

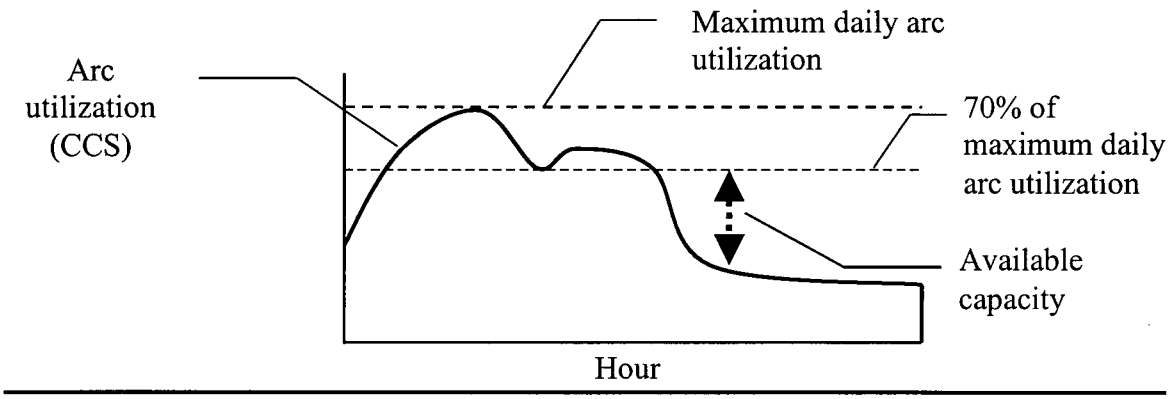


Figure 11 Calculation of available capacity

Since we are using maximum daily arc utilization instead of using the actual capacity, this calculation provided a lower bound to the available capacity. However, the maximum daily arc utilization should not be too far from the actual arc capacity since arcs are designed such that there is overflow. Therefore, the maximum arc utilization should be fairly close to the actual capacity. The available capacity for the new routes was calculated by taking the minimum of the available capacity of the two arcs composing that route.

Once this hourly available capacity was calculated for each route, we then tried to determine which routes had sufficient available capacity to be used as new alternate routes. The following criterion was used to determine this: If there is sufficient available capacity on the route to carry 5% of the maximum daily arc utilization on the AZ pair of interest, then this route can be used as an alternate route. This value, 5%, can be increased depending on the percentage of blocked calls (i.e. calls that could not be made because all of the circuits on all of the possible routes were busy) that are noticed.

4 Results and Analysis

4.1 Results of the Principal Components Analysis

4.1.1 Description of the results

The principal components can either be calculated from the correlation matrix or from the variance-covariance matrix. Since the data used for the principal component analysis had been normalized, the PCA was first performed using the covariance-variance matrix. The results obtained from the PCA are shown in Figures 12 to 14. The first figure, Figure 12, shows the mean and the standard deviation for each of the 16 variables (the normalized arc utilization on an hourly basis for hours 7:00 to 22:00).

The table in Figure 13 has four columns containing the PCA results. The first column lists the eigenvalues. The second column indicates the difference between two successive eigenvalues. For example, the difference between the first and the second eigenvalue is 0.37805360 - 0.11056362, which is equal to 0.26748998. The third column indicates the amount of variance accounted for by each eigenvalue. For example, the proportion of the variance that is accounted for by the first eigenvalue is equal to

$$\lambda_1/tr(\Sigma) = \lambda_1/(\lambda_1 + \lambda_2 + \ldots + \lambda_{16}) = 0.378054/0.595029$$

Therefore, $\lambda_1/tr(\Sigma) = 0.6354$.

The last column in Figure 13 indicates the cumulative variance that has been explained by the eigenvalues. For example, the proportion of the variance that was accounted for by the three first eigenvalues is equal to:

$$\lambda_1/tr(\Sigma) + \lambda_2/tr(\Sigma) + \lambda_3/tr(\Sigma) = 0.6354 + 0.1858 + 0.0504 = 0.8716$$

Figure 14 shows the elements of the five first principal components. For example, the first principal component (or eigenvector a_1) is equal to:

$$a_1 = [-0.024492,\ -0.063914, -0.088346, \ldots, 0.511045, 0.391758]^T$$

It can be verified that every eigenvector is normalized to have a length of 1 by taking the square root of the sum of squares of each element that compose an eigenvector. For example, the length of the first eigenvector is equal to:

$$\text{Length of } a_1 = \sqrt{(-0.024492)2 + (-0.063914)2 + \ldots + (0.391758)2} = 1$$

Finally, Figure 15 shows the principal component scores for the first three components of the first observation (the complete table includes the 16 principal component scores for each of the 458 observations). For example, the first principal component score for the AZ pair 00-01 is equal to –1.308. This can be calculated as follows:

$y_{11} = a_1(x_1 - \mu)$, where

$a_1 = [-0.024492,\ -0.063914, -0.088346, \ldots, 0.511045, 0.391758]^T$
$x_1 = [0.167, 0.521, 0.919, \ldots, 0.030, 0.000]^T$, and
$\mu = [0.007730199, 0.349277637, 0.772263555, \ldots, 0.66555774, 0.457278907]^T$

Therefore,

$y_{11} = -0.024492\ (0.167 - 0.007730199) + \ldots + 0.391758(0.000 - 0.457278907) = -1.308$

Observations 458
Variables 16

Simple Statistics

	N7	N8	N9	N10	N11	N12
Mean	0.007730199	0.349277637	0.772263555	0.911521617	0.88417	0.72166
StD	0.028739121	0.105364623	0.132685249	0.118782922	0.13343	0.14284

	N13	N14	N15	N16	N17	N18
Mean	0.727745071	0.741304534	0.738477416	0.702182757	0.54789	0.5174
StD	0.144098771	0.151025581	0.146209116	0.151097151	0.19714	0.22158

	N19	N20	N21	N22
Mean	0.566095759	0.631980559	0.66555774	0.457278907
StD	0.262016675	0.295733564	0.323691912	0.276683323

Figure 12 Simple statistics obtained from the PCA

The PRINCOMP Procedure

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	0.378054	0.267490	0.6354	0.6354
2	0.110564	0.080563	0.1858	0.8212
3	0.030000	0.016168	0.0504	0.8716
4	0.013833	0.004444	0.0232	0.8948
5	0.009388	0.001070	0.0158	0.9106
6	0.008319	0.000438	0.0140	0.9246
7	0.007881	0.001988	0.0132	0.9378
8	0.005892	0.000349	0.0099	0.9477
9	0.005543	0.000482	0.0093	0.9571
10	0.005062	0.000287	0.0085	0.9656
11	0.004775	0.000408	0.0080	0.9736
12	0.004366	0.000484	0.0073	0.9809
13	0.003882	0.000374	0.0065	0.9874
14	0.003508	0.000096	0.0059	0.9933
15	0.003412	0.002863	0.0057	0.9991
16	0.000549	0.000900	1.0000	

Figure 13 Eigenvalues of the Covariance Matrix

Eigenvectors

	Prin1	Prin2	Prin3	Prin4	Prin5
N7	-0.024492	-0.004037	-0.010826	0.028389	-0.032183
N8	-0.063914	0.134548	0.205979	0.349622	-0.132616
N9	-0.088346	0.200605	0.335312	0.510797	0.352564
N10	-0.069634	0.227641	0.130876	0.105001	0.547417
N11	-0.068662	0.314405	0.001504	0.017672	0.124444
N12	0.002172	0.380914	-0.073266	0.043562	-0.027391
N13	-0.023642	0.373949	-0.136439	0.207184	-0.216122
N14	-0.046425	0.395692	-0.143375	0.042403	-0.38735
N15	-0.008874	0.3842	-0.118786	-0.041066	-0.191523
N16	0.120626	0.325399	-0.004167	-0.255443	0.099482
N17	0.265262	0.24284	0.021992	-0.335453	0.275305
N18	0.323978	0.149997	0.153356	-0.402066	0.166741
N19	0.400009	0.05185	0.338255	-0.06976	-0.284968
N20	0.458938	-0.040475	0.326169	0.23018	-0.256801
N21	0.511045	-0.096269	-0.03053	0.289206	0.140569
N22	0.391758	-0.030057	-0.723524	0.274726	0.165343

Figure 14 Five first Principal Components

Obs	AZ_ID	N7	N8	N9	N10	N11	N12	N13	N14	N15
1	00-01	0.167	0.521	0.919	0.966	1.000	0.720	0.798	0.852	0.799

N16	N17	N18	N19	N20	N21	N22	PRIN1	PRIN2	PRIN3
0.509	0.166	0.071	0.038	0.030	0.030	0.000	-1.308	0.043	-0.043

Figure 15 Principal Component Scores first the first observation

4.1.2 How many principal components should be retained?

We determined how many principal components we needed to keep by looking at the SCREE plot and by looking at how much variance each principal components accounted for. The principal component analysis was also repeated using the correlation matrix.

The result of the PCA using the correlation matrix (instead of the variance-covariance matrix) is shown in Appendix C .

Once again, we looked at the three criteria in order to determine how many principal components to keep. This time, when looking at the eigenvalues (see Appendix C), we found that the three first eigenvalues were greater than one. This suggested that three principal components should be kept. Then, we looked at the proportion of the variance explained by each of the eigenvalues. From this criterion, we decided to keep the three first principal components as the three first eigenvalues each capture more than 5% of the variance.

Finally, from the SCREE plot of the eigenvalues, Figure 16, we could see that value of the eigenvalues tends to level off after the third eigenvalue. This suggests that the dimensionality of the problem can probably be reduced to 3: principal component 1 (PRIN1), principal component 2 (PRIN2) and principal component 3 (PRIN3).

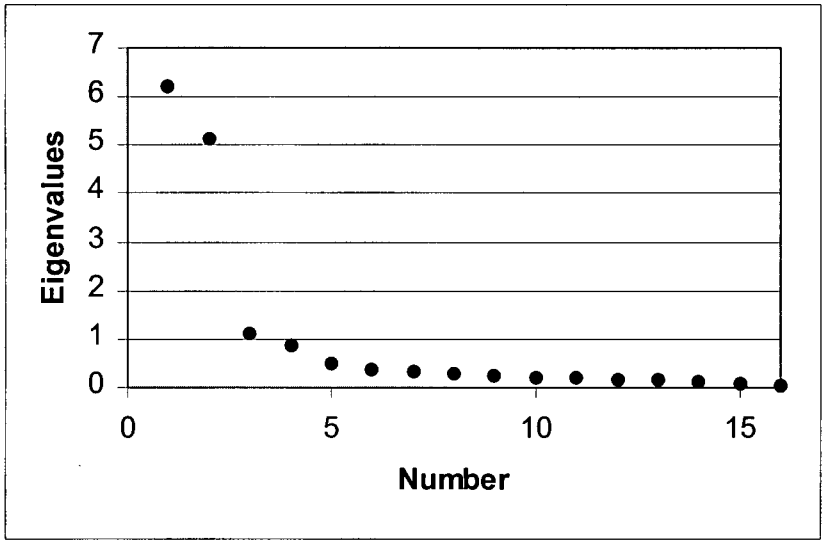


Figure 16 SCREE plot of eigenvalues

Since all three criteria were suggesting that we should keep the first three components, we pursued the analysis only using these new variables instead of using the 16 original variables.

4.1.3 Screening the data using PCA

The scatterplots of PRIN1 vs. PRIN2, PRIN2 vs. PRIN3 and PRIN2 vs. PRIN3 were then plotted to screen the data and determine if there are any outliers (see Appendix D). From the graph of PRIN2 vs. PRIN3 and PRIN2 vs. PRIN3 it was apparent that there were two outliers. These outliers correspond to the demand on AZ pairs 01-04 and 11-27. The reason why these two AZ pairs were outliers was because they had an unusually high morning demand (see Appendix E). These were removed from the data before performing the next cluster analysis so they would not influence the statistics[4]. The PRINCOMP procedure was run with the new data (without the two outliers) and once again, the scatterplots were graphed. This time, there were no apparent outliers. There were no easily identified clusters on these new scatterplots so another method was used to group the AZ pairs. However, the 3 principal components scores obtained from the principal components analysis were still useful as these new variables can be used instead of the 16 original variables.

4.1.4 Interpretation of the principal components

The first three principal components are shown in Figure 17. These components can be interpreted as follows:

- **Component 1**: Weighs highly the AZ pairs which have very low arc utilization in the morning and in the afternoon and a high utilization in the evening.

[4] The analysis was also performed keeping the two outliers. These outliers were detected by the cluster analysis: they were grouped together, with no other observations. This did not form a useful cluster. It is therefore better to first screen the data using the PCA, then to remove the outliers and continue the analysis without the outliers.

- **Component 2**: Weighs highly AZ pairs that have high arc utilization in the afternoon and low arc utilization in the morning and in the afternoon.

- **Component 3**: Weighs highly AZ pairs that have high arc utilization in the morning, low arc utilization in the afternoon and moderate arc utilization in the evening.

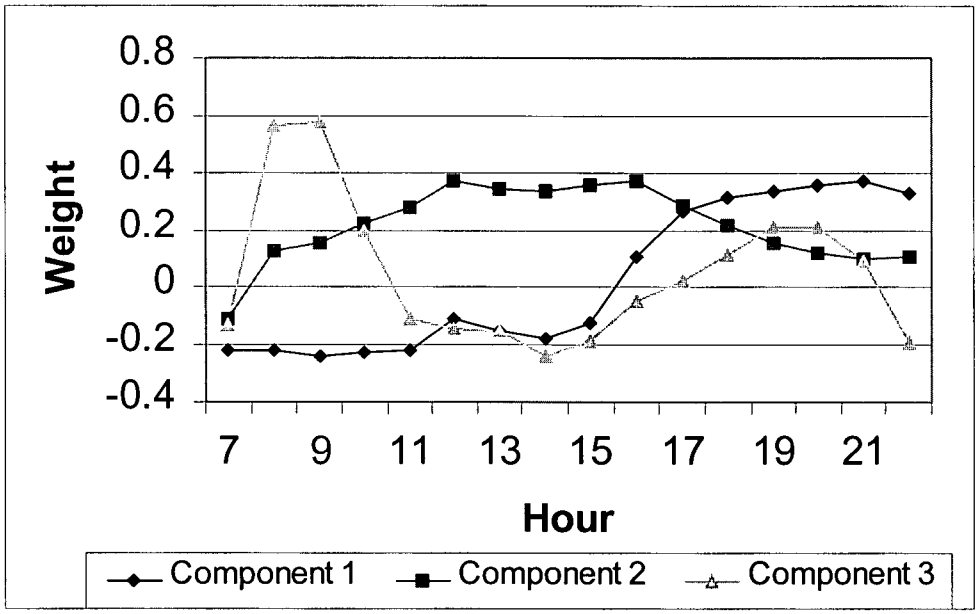


Figure 17 Graphical representation of first three principal components

Based on these interpretations of the principal components, we would expect that AZ pairs which have demand patterns similar to the pattern of the weights given to each hour by a certain principal component to have a high score for that component. These interpretations are illustrated in the following examples.

Example 1: The demand for the AZ pair 02-29 is shown in Figure 18. From this time series of the normalized arc utilization, we can see that the demand in the morning is relatively low compared to that of the afternoon and that the demand in the evening is relatively high compared to the demand in the afternoon. Based on the interpretations described above, we can expect that the first principal component score will be very high, the second principal component score will be very low and that the third principal component score will be very low. This was the case. The first principal component

scores range from -6.0584 to 4.83846 and this AZ pair obtained a score of 4.68528, which is a high score and means that it shape resembles that of the principal component one. The second principal component scores range from -8.66150 to 5.61533 and this AZ pair obtained a score of -6.30489. Finally, the third component scores range from -2.55006 to 3.2398 and this AZ pair obtained a score of -2.34666.

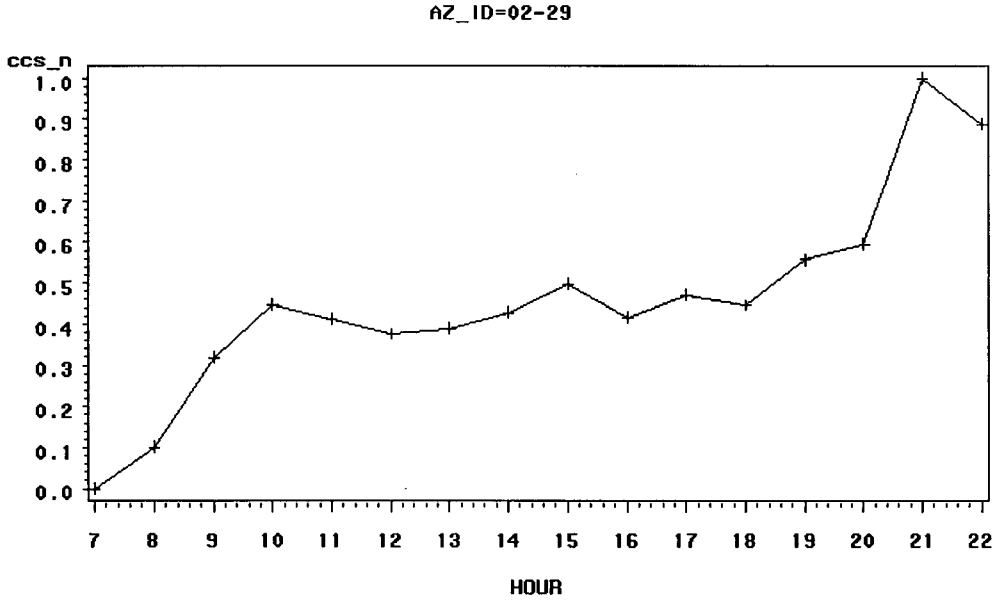


Figure 18 Example 1: Interpretation of the principal components

Example 2: The AZ pair shown in Figure 19 shows a high demand in the morning relatively to a lower demand in the afternoon and a very low demand in the evening. We can therefore expect that the score for the first principal component will be very low, as this pattern is exactly the opposite of that depicted by principal component one. The second principal component score will be low and the third principal component will also be low, as the demand in the evening is not high. This AZ pair obtained a score of -5.56782 for the first principal component, -1.15865 for the second principal component and -0.8837 for the third one.

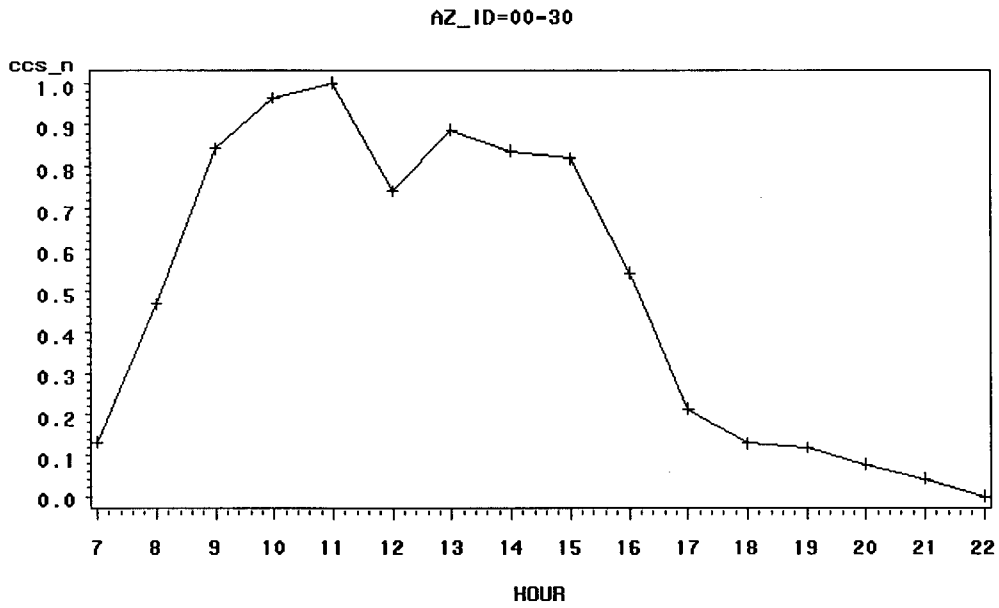


Figure 19 Example 2: Interpretation of the principal components

4.2 Results of the cluster analysis

The clustering was first performed using the FASTCLUS procedure. This analysis was done on the three first principal components. The clusters obtained from using this procedure are shown in Appendix F . This procedure seemed to give good results as the clusters were sufficiently large (none of the clusters had small clusters with only 1 or 2 observations). The main concern that we had with this procedure is that it is very dependent on the user-specified number of clusters and on the selection of the initial set of seed points. We did not know from using this clustering procedure what would be an appropriate number of clusters. We therefore looked at hierarchical clustering methods, since they have criteria that can be used to help determine how many clusters should be kept. We used the CLUSTER procedure in SAS and then selected the clustering method.

We first performed the clustering using the average linkage method. The cluster analysis was performed using the first 3 principal component scores[5]. Figure 20 shows the output for the 15 first clusters of the cluster analysis. Each row represents the statistics calculated after two clusters have joined. The first column, NCL, represents the number of clusters left. In the complete table, in the first row NCL would be equal to the total number of observations minus 1. The two next columns show which clusters have joined during this step. For example, when the number of clusters was 11, the cluster 240 (i.e. a group of AZ pairs that was renamed cluster 240) was merged with the AZ pair 07-27 (this AZ pair was still in a cluster by itself before this step). The next column shows the frequency or number of observations in each group. Finally, the rest of the columns represent statistics calculated on these clusters.

In order to determine the appropriate number of clusters, the Cubic Clustering Criterion (CCC), the Pseudo Hotelling's T^2 Test (PST2) and the Pseudo F statistics (PSF) results were examined (see Figure 20). These criteria are explained more in detail in the background section.

As it was mentioned earlier in this thesis, when the value of PST2 is small, this indicates that the clusters should be combined. From looking at the values of PST2 in Figure 20, we can see that this statistic indicates that the appropriate number of clusters should be either 5, 9 or 12 since the PST2 values for these mergers are relatively small compared to the others.

[5] We also attempted to perform the cluster analysis directly on the raw data (instead of on the principal components retained), but these clusters were less interpretable. Therefore, we continued out analysis using the principal components as inputs to the cluster analysis. See

The CCC statistic is not very helpful in this case since only two of the values are positive and none of them of greater than three. Therefore, the CCC statistic cannot be used as an indication of the number of clusters to use.

Another criterion that can be used to judge the number of clusters in a data set is to look at the pseudo *F* statistic (PSF). Relatively large values indicate that the number of clusters is probably appropriate. From reading the PSF values in Figure 20, we can see that this criterion indicates that the ideal number of clusters is possibly 2, 9 or 11.

From these three criteria, it seems that the number of clusters should be between 2 and 9 (forming more than 9 clusters is not very useful for this analysis as we are trying to simplify the data).

Average Linkage Cluster Analysis

Cluster History

NCL	- - Clusters	Joined---	FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Norm RMS Dist	T i e
15	CL32	CL85	48	0.0029	0.88	0.904	-6.4	233	16.7	0.487	
14	CL15	CL52	56	0.0049	0.875	0.899	-6.2	240	22.4	0.49	
13	CL14	CL24	109	0.0181	0.857	0.893	-8.7	223	71.4	0.515	
12	CL20	CL59	6	0.0012	0.856	0.887	-7.2	241	4.1	0.53	
11	CL240	07-27	3	0.0009	0.855	0.879	-5.6	264	36.8	0.557	
10	CL17	CL18	177	0.0312	0.824	0.871	-9.6	233	94.1	0.56	
9	CL16	CL33	51	0.0042	0.82	0.861	-8.1	256	12.6	0.56	
8	CL22	CL10	231	0.0385	0.781	0.849	-12	230	85.6	0.622	
7	CL27	CL9	62	0.0108	0.771	0.834	-11	252	28.5	0.642	
6	CL19	CL44	47	0.006	0.765	0.815	-6.6	294	46.9	0.653	
5	CL11	CL12	9	0.0026	0.762	0.788	-3.3	363	5.6	0.678	
4	CL6	CL13	156	0.0546	0.707	0.748	-4.6	366	147	0.729	
3	CL7	CL5	71	0.0126	0.695	0.682	1.4	518	22.1	0.8	
2	CL8	CL3	302	0.1158	0.579	0.548	2.2	627	179	0.891	
1	CL4	CL2	458	0.579	0	0	0	.	627	1.298	

Figure 20 Results of the cluster analysis using Average Linkage Method

Another way of determining the optimal number of clusters is to look at the tree generated from the cluster analysis. This tree is shown in Figure 21. From this tree, it seems that there are three main clusters. These are then subdivided to form a total of 7 or

8 clusters (two of which include only a few observations). This value is within the range found using the three criteria described above. Therefore, we continued the analysis using 7 clusters. These 7 clusters are shown in Appendix G . Each graph represents the box plot of the normalized time series of the AZ pairs belonging to that cluster.

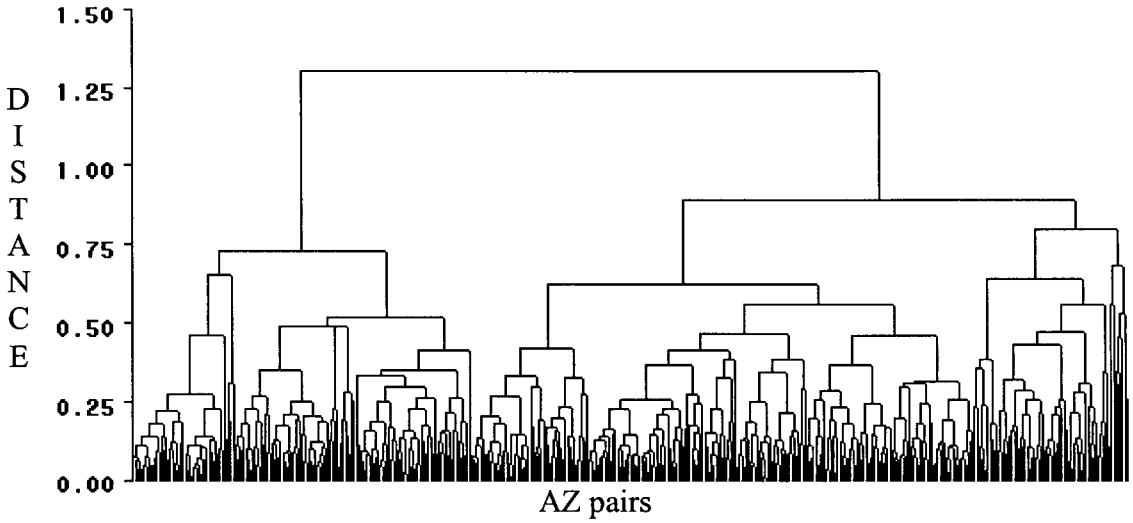


Figure 21 Tree diagram obtained from using the Average Linkage Method

Which hierarchical method should be used?

We repeated this cluster analysis using the Ward's and the Centroid clustering methods. The clusters obtained from these methods are shown in Appendix H and Appendix I . The Centroid Method tended to identify outliers and then group by themselves. The Average Linkage Method also tended to form small clusters (see clusters 5 and 6). The clusters obtained using Ward's Minimum Variance Method seemed more reasonable. The size of the clusters seemed relatively similar. This is probably the group sizes that are the most useful for the rest of the analysis. Therefore, we continued the analysis using Ward's method. This result is similar to that obtained by Puterman and Dunn (1986) in their cluster analysis on data concerning children with low birthweight. In their

study, they compared several hierarchical methods, including nearest neighbour, farthest neighbour, centroid, median and Ward's methods. They found that the results obtained from using Ward's method corresponded the most closely with their objective.

4.3 Results of the search for adjacent arcs

The groups found from using the cluster analysis were then used to find adjacent arcs. From using the methodology described in section 3.6 and illustrated in Figure 10, we were able to find new alternate routes that can be used during certain periods of the day. For this thesis, the methodology was used to find if there were possible alternate routes in each of the other groups. For example, for each of the AZ pairs in the first group, we searched for possible alternate routes in groups 2 to 7. The table shown in Appendix J shows a sample of the resulting new routes. This table summarizes which nodes can used as tandem for a certain AZ pair. In addition, it indicates when these nodes can be used as tandems ("Y" indicates that the node can be used as a tandem during that hour and "N" indicates that it cannot be used during that hour). For instance, in the example shown in Table 7, the first line indicates that the node 00 could be used as a tandem between 7AM and 9AM and between 4PM to 23PM for the AZ pair 02-05. This new alternative route would direct calls from the origin switch 02, through the tandem switch 00 and finally to the destination 05. The next line shows that node 01 could also be used as a tandem for the AZ pair 02-05 during these hours.

AZ_id	Tandem	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	H21	H22
02-05	00	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
02-05	01	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y

Table 7 Example of the resulting new possible alternate routes

However, in this example, the node 00 should not be used as a tandem during the time period between 9AM and 4PM since there is either not sufficient available capacity on

the arc 02-00 or 00-05. Similarly, there is not sufficient available capacity on either the arc 02-01 or 01-05 for this route to be used as an alternate route between 9AM and 4PM.

4.4 How will these routes be used?

All the new alternate routes that were found do not take into account the possibility that a same arc could be used as part of more than one new alternate route. For example, the arc 05-06 could be part of the new alternate route 05-06-09 for the AZ pair 05-09 and it could also be listed as the new alternate route 01-05-06 for the AZ pair 01-06. The reason for this is that the work done in this thesis was to generate all of the possible routes to see if there was enough potential to use this methodology to find new routes and then implement them. Therefore, it would not be possible to implement them all of these new routes at once. The user would rather add only a few the possible routes, making sure that the these new routes do not have any arcs in common. This routes would be added to the routing table during the periods when they have sufficient capacity. For example, based on the example shown in Table 7, if the user noticed that a large number of the calls going from the switch 02 to the switch 05 were blocked between 4PM and 6PM, then either the route 02-00-05 or 02-01-05 could be added to the routing table during these hours. Based on these results and our methodology, we have listed a few recommendations.

5 Recommendations

The results show that it is possible to group the demand patterns on the arcs using cluster analysis and that is possible to find routes based on those groups that have a demand pattern complementary to that of a specified AZ pair. However, the results obtained for this thesis are based on one day of data, which was all that was available at the time of

this study. Therefore, once data is available for other days, the methodology described in this thesis should be applied in order to see how much the results vary from one day to another. In addition, once data is available regarding the actual arc capacities, these values should be used, instead of the maximum arc utilization, to calculate the available capacity on each arc.

Once more data is available, the analysis described above should be performed on this new data to determine if the number of significant principal components and if the number of useful clusters vary from one day to the other. If it does not, then the methodology described in this thesis could be automated.

In addition, the code could be adjusted so that instead of searching for all of the possible new alternate routes, the code only searched for possible alternate routes for a specific AZ pair. This *adjusted* code could be incorporated into a tool (with a visual interface) that could be used by Telus. This tool would be very useful to Telus since if they start to see blocked calls on a certain AZ pair, they could use this tool and determine if there are any potential alternate routes that could carry extra traffic. This would permit Telus to avoid or delay adding capacity to that AZ pair.

References

Braun, D., *Efficient routing of telephone calls in a circuit-switched network,* unpublished M.Sc. thesis, University of British Columbia, Faculty of Commerce and Business Administration, 2000.

Cook, D., *Principal Component Analysis*, November 23, 2000, http://www.public.iastate.edu/~dicook/stat501/97/lectures/2.6.html

Hatcher, L., Stepanski, E. J., *A Step-by-Step Approach to Using the SAS® System for Univariate and Multivariate Statistics*, Cary, NC:SAS Institute Inc. 1994, 552 pp.

Johnson, D. E., Applied Multivariate Methods for Data Analysts, Duxbury Press, 1998.

Puterman, M. L., Dunn, H. G., 'Statistical Analysis of Mild Brain Dysfunctions', *In:* Henry G. Dunn (Ed.), *Sequelae of Low Birthweight: The Vancouver Study*, London: Mac Keith Press, 1986, pp.114-125.

SAS Institute Inc., *SAS/STAT® User's Guide, Version 6, Fourth Edition*, Volume 1, Cary, NC: SAS Institute Inc., 1989. 943 pp.

SAS Institute Inc., *SAS/STAT® User's Guide, Version 6, Fourth Edition*, Volume 2, Cary, NC: SAS Institute Inc., 1989. 846 pp.

SAS Institute Inc., *SAS® User's Guide:Basics, Version 5 Edition,* Cary, NC: SAS Institute Inc.,1985. 1290 pp.

SAS Institute Inc., *SAS® Macro Language: Reference, First Edition*, Cary, NC: SAS Institute Inc., 1997. 304 pp.

Appendices

Appendix A Description of some SAS procedures

A-1 Principal Components Analysis using the SAS PRINCOMP procedure

The procedure PRINCOMP in SAS can be used to perform PCA. The syntax for this procedure is the following[6]:

PROC PRINCOMP <options>; required statement

BY variables;
FREQ variables;
PARTIAL variables;
VAR variables;
WEIGHT variables;
} optional statements

In our analysis, we have used the following options:

- COVARIANCE (COV): indicates that the principal components should be computed from the covariance matrix. If this option had not been specified, the principal components would have been calculated using the correlation matrix. The COV option should not be used unless the units for each variable are comparable and the variables have been standardized in some way. The variables in this data set are all measured in centi-call seconds (CCS) and they have been normalized. Therefore, it is possible to use the COV option.
- DATA: indicates which data set will be analyzed.
- OUT: creates a data set that contains, in addition to all of the original data, the principal component scores.
- OUTSTAT: creates a data set that contains the means, standard deviations, number of observations, correlations or covariances, eigenvalues and eigenvectors.

We have only used the following optional statement:
- VAR: lists the variables that will be analyzed.

[6] More information about this procedure can be found in SAS/STAT® User's Guide, volume 2, pp.1243 to 1263.

A-2 Cluster analysis using the SAS FASTCLUS procedure

The procedure FASTCLUS in SAS can be used to perform cluster analysis. The syntax for this procedure is the following[7]:

PROC FASTCLUS MAXCLUSTERS=n RADIUS=t <options>	required statement
VAR variables; ID variables; FREQ variables; WEIGHT variables; BY variables;	optional statements

As part of the required statement, either the maximum number of clusters allowed (MAXCLUS) or the minimum distance allowed between each cluster seeds (RADIUS) must be specified.

In our analysis, we have used the following options:

- MAXITER: indicates the maximum number of iterations allowed for recomputing the cluster seeds
- DATA: indicates the name of the data set containing the data to be clustered
- OUT: creates a new data set that contains, in addition to all of the original data, the variables CLUSTER and DISTANCE

We have also used the following optional statement:

- VAR: indicates which variables to use in the cluster analysis.

[7] More information about this procedure can be found in SAS/STAT® User's Guide, volume 1, pp.825 to 850.

A-3 Cluster analysis using the SAS CLUSTER procedure

The procedure CLUSTER in SAS can be used to perform cluster analysis. The syntax for this procedure is the following[8]:

PROC CLUSTER METHOD=name <options>;	required statement
BY variables; COPY variables; ID variables; RMSSTD variables; VAR variables;	optional statements
FREQ variables;	required when RMSSTD statement is used, otherwise it is optional

As part of the required statement, the clustering method to be used needs to be specified. The methods available in SAS are: Average, Centroid, Complete, Density, EML, Flexible, McQuitty, Median, Single, Twostage and Ward. These clustering methods are described in more detail in the background section of this thesis.

In our analysis, we have used the following options:

- DATA: indicates the name of the data set containing the data to be clustered.
- CCC: prints the Cubic Clustering Criterion. This option should not be used in conjunction with the METHOD=Single.
- OUTTREE: creates a new data set that can be used by the TREE procedure to draw a tree diagram.
- PSEUDO: prints pseudo F and t^2 statistics. This option is effective only when the data are coordinates or METHOD=Average, Centroid or Ward. This option should not be used in conjunction with the METHOD=Single.

We have also used the following optional statement:

- VAR: indicates which variables to use in the cluster analysis.

[8] More information about this procedure can be found in SAS/STAT® User's Guide, volume 1, pp.519 to 614.

A-4 Tree diagram using the SAS TREE procedure

The procedure TREE in SAS can be used to print tree diagrams. This can be used to view how clusters were formed at different levels. The syntax for this procedure is the following[9]:

PROC TREE <options>;	required statement
NAME variables; HEIGHT variables; PARENT variables; BY variables; COPY variables; FREQ variables; ID variables;	optional statements

In our analysis, we have used the following options:

- DATA: indicates the name of the data set containing the data used to define the tree.
- OUT: creates a new data that lists all of the observations in the tree, as well variables describing to which clusters each observation belongs to. If this option is used, either NCLUSTERS or LEVEL options must be used.
- NCLUSTERS: Specifies the number of clusters to keep.

We have also used the following optional statement:

- COPY: indicates which variables to copy to the output file.
- ID: indicates which variable should be used to identify the objects in the tree in the output file.

[9] More information about this procedure can be found in SAS/STAT® User's Guide, volume 2, pp.1613 to 1631.

Appendix B Results of the PCA performed on non-normalized data

The results of this analysis can be summarized as follows. The SCREE plot showed that only one principal component should be used. This principal component attributed basically the same weight to each variables. Each element of the first eigenvector was around 0.25, which corresponded to giving an equal weight to each variable. The values of 0.25 can be calculated as follows: the 16 elements (representing hours 7 through 22) were given a similar magnitude therefore each of them has a value of $\pm 1/\sqrt{16} = \pm 0.25$. Thus, this principal component only captured the variance due to the volume of calls, not the pattern of the variation of the demand throughout the day. This supports using normalized data for the analysis.

	Eigenvalue	Difference	Proportion	Cumulative
PRIN1	15.3705	14.9277	0.960657	0.96066
PRIN2	0.4428	0.3305	0.027674	0.98833
PRIN3	0.1123	0.0797	0.007019	0.99535
PRIN4	0.0326	0.0183	0.002037	0.99739
PRIN5	0.0143	0.0086	0.000896	0.99828

Figure 22 Eigenvalues of the correlation matrix

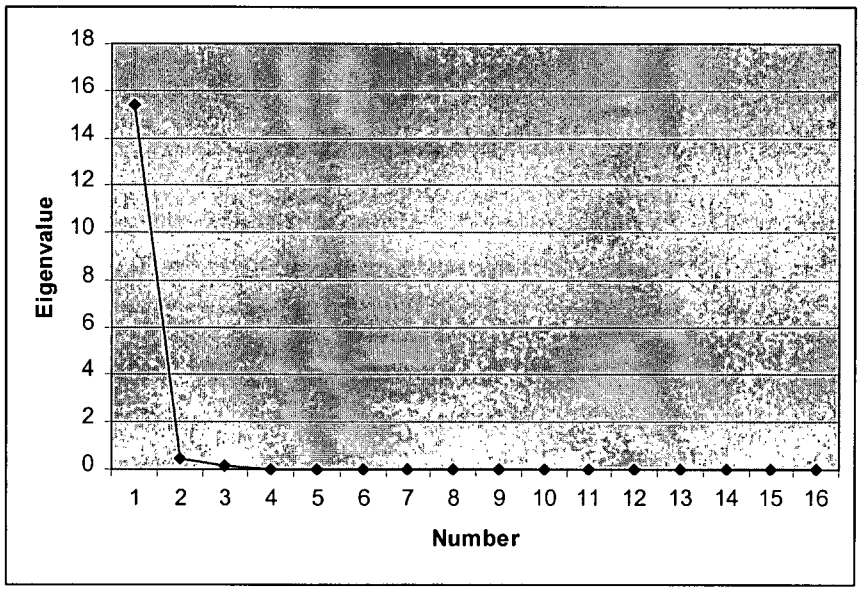


Figure 23 SCREE plot of the eigenvalues

	PRIN1	PRIN2	PRIN3
7	0.241845	-0.266401	0.772399
8	0.247786	-0.307951	0.281299
9	0.250951	-0.239003	-0.142073
10	0.251656	-0.193613	-0.248267
11	0.251852	-0.193085	-0.237668
12	0.253404	-0.141858	-0.138891
13	0.252909	-0.172962	-0.084318
14	0.25269	-0.175234	-0.119929
15	0.253603	-0.125743	-0.116743
16	0.25443	-0.017332	-0.136451
17	0.25285	0.160231	-0.095562
18	0.249955	0.273859	-0.061689
19	0.247392	0.349123	-0.004427
20	0.246958	0.361664	0.015152
21	0.245921	0.386733	0.053651
22	0.245398	0.320195	0.311235

Figure 24 First three normalized eigenvectors, a_1, a_2, a_3

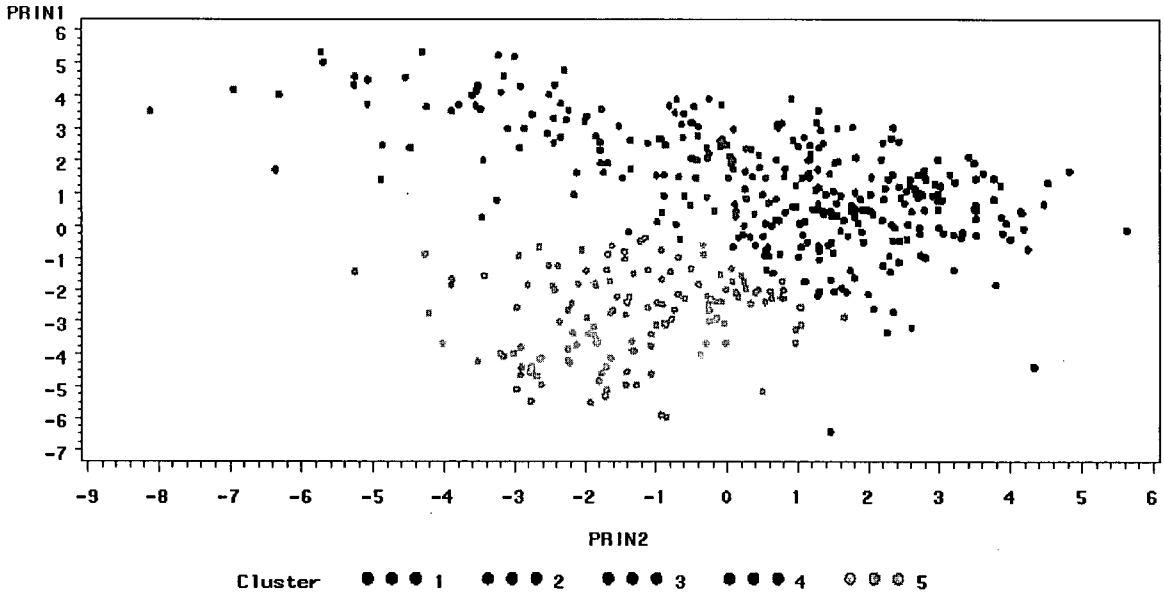
Appendix C Results of the PCA using the correlation matrix

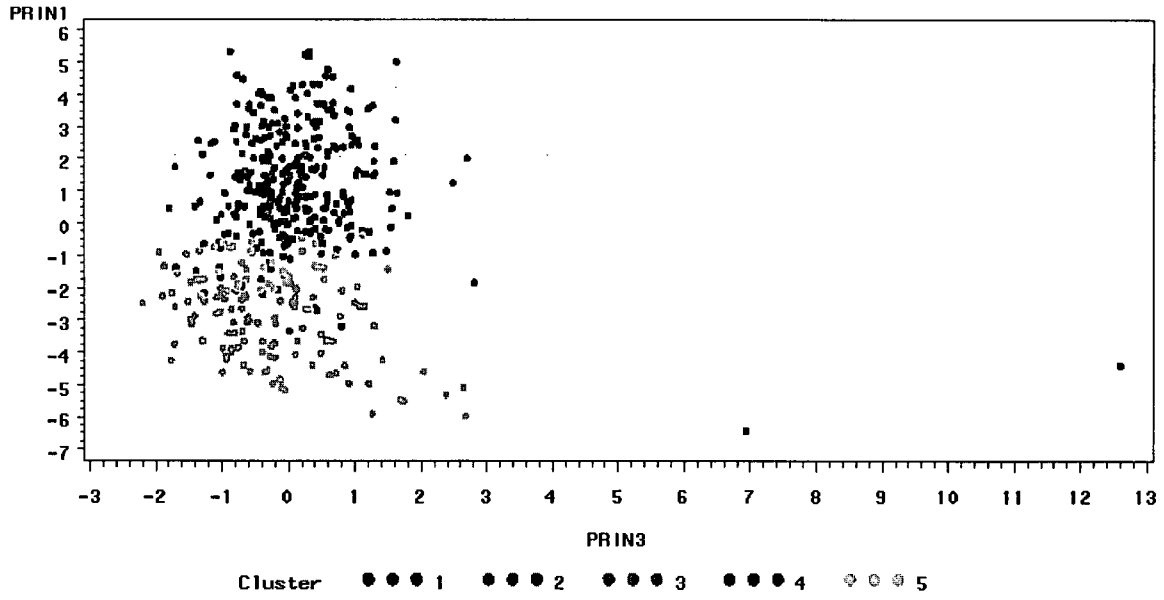
Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	6.32892724	1.16051626	0.3956	0.3956
2	5.16841098	4.1627288	0.323	0.7186
3	1.00568218	0.23363881	0.0629	0.7814
4	0.77204337	0.24657065	0.0483	0.8297
5	0.52547272	0.15731555	0.0328	0.8625
6	0.36815718	0.03922067	0.023	0.8855

	Prin1	Prin2	Prin3	Prin4	Prin5
N7	-0.218068	-0.107553	-0.131231	0.767449	0.53781
N8	-0.221336	0.127624	0.562158	0.370364	-0.441
N9	-0.242742	0.156286	0.579287	-0.025605	0.13178
N10	-0.225897	0.221177	0.193285	-0.396466	0.53323
N11	-0.221985	0.282967	-0.107083	-0.092848	0.25468
N12	-0.107069	0.371136	-0.140696	-0.002884	0.02445
N13	-0.15081	0.344498	-0.152119	0.167608	-0.1139
N14	-0.178797	0.333567	-0.239056	0.089484	-0.198
N15	-0.119134	0.357717	-0.183128	0.047417	-0.2091
N16	0.107078	0.367847	-0.046361	-0.058797	-0.041
N17	0.266625	0.284289	0.026095	0.008205	0.10824
N18	0.313069	0.216859	0.113953	0.050463	0.12271
N19	0.337932	0.158114	0.210605	0.115269	0.09699
N20	0.354248	0.116765	0.210209	0.090069	0.07887
N21	0.367845	0.097499	0.090043	0.110395	0.09446
N22	0.329076	0.108541	-0.190454	0.171397	-0.0139

Appendix D Scatterplots of the first three principal components





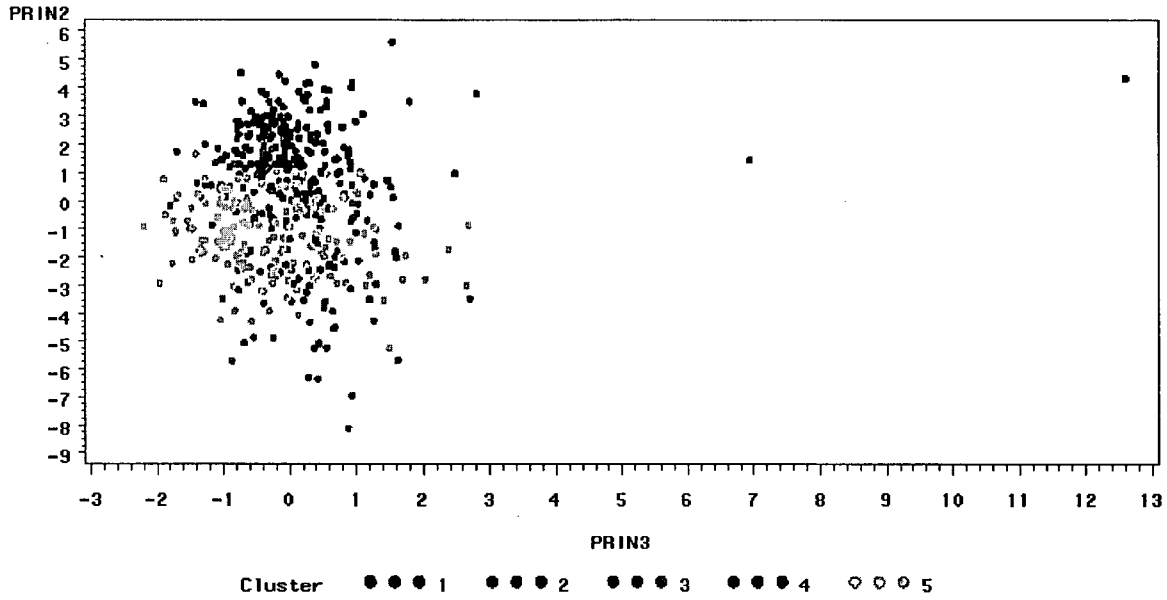
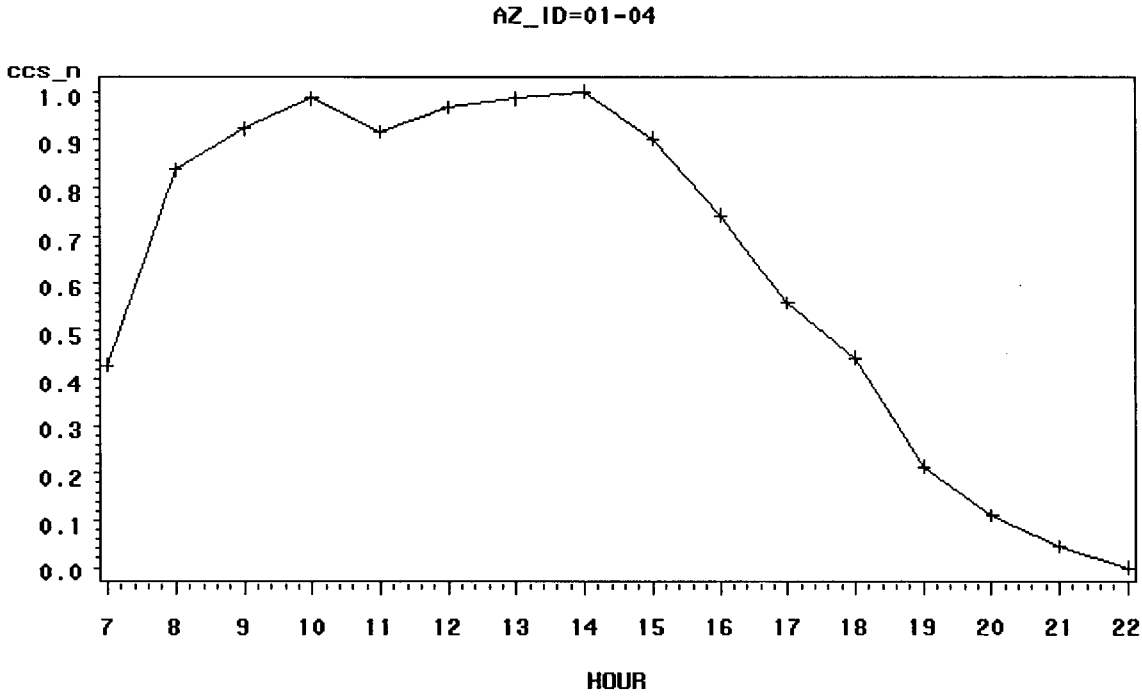
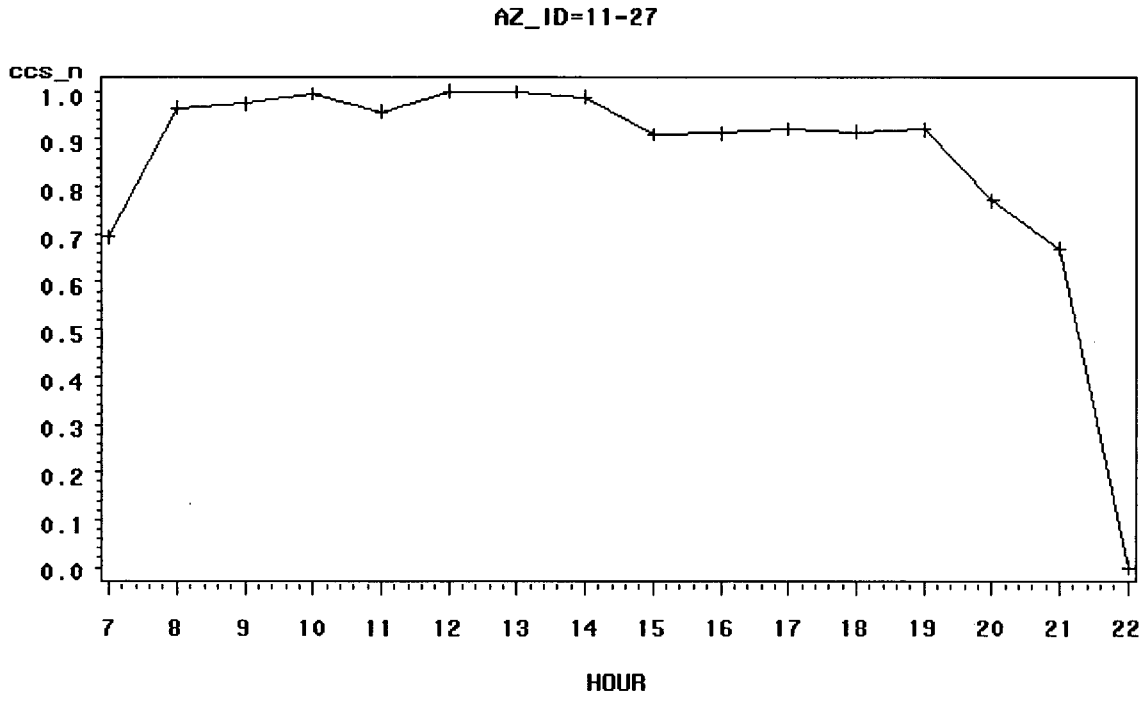


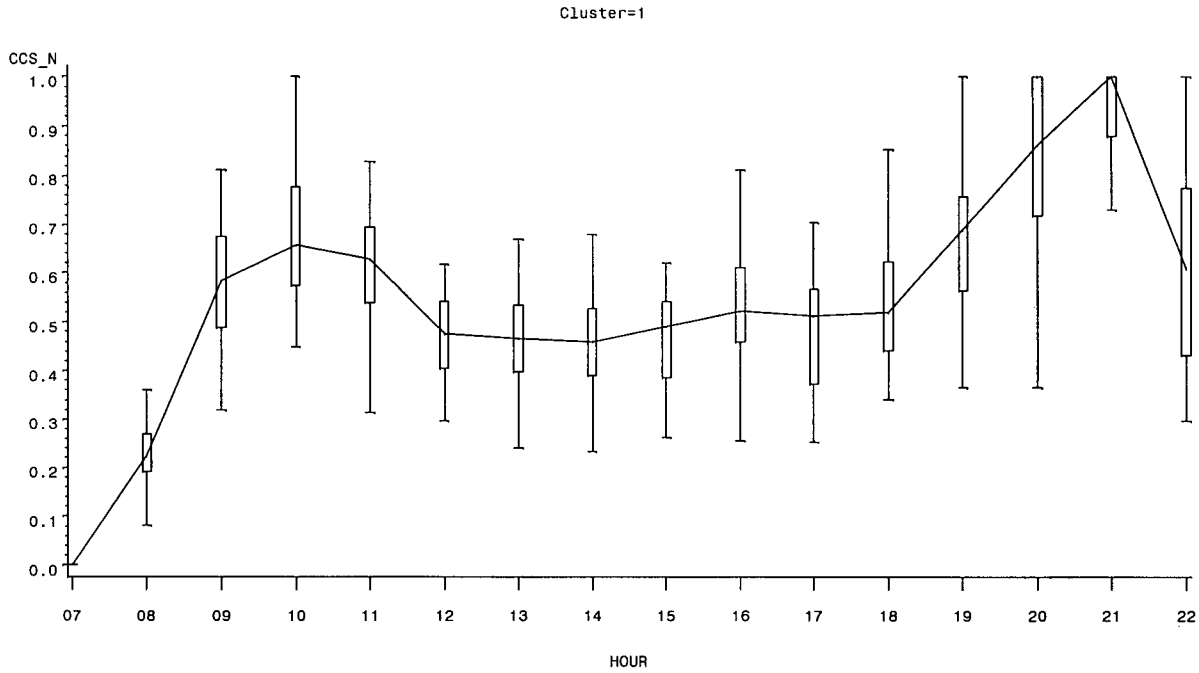
Figure 25 Scatterplots of the 3 first components

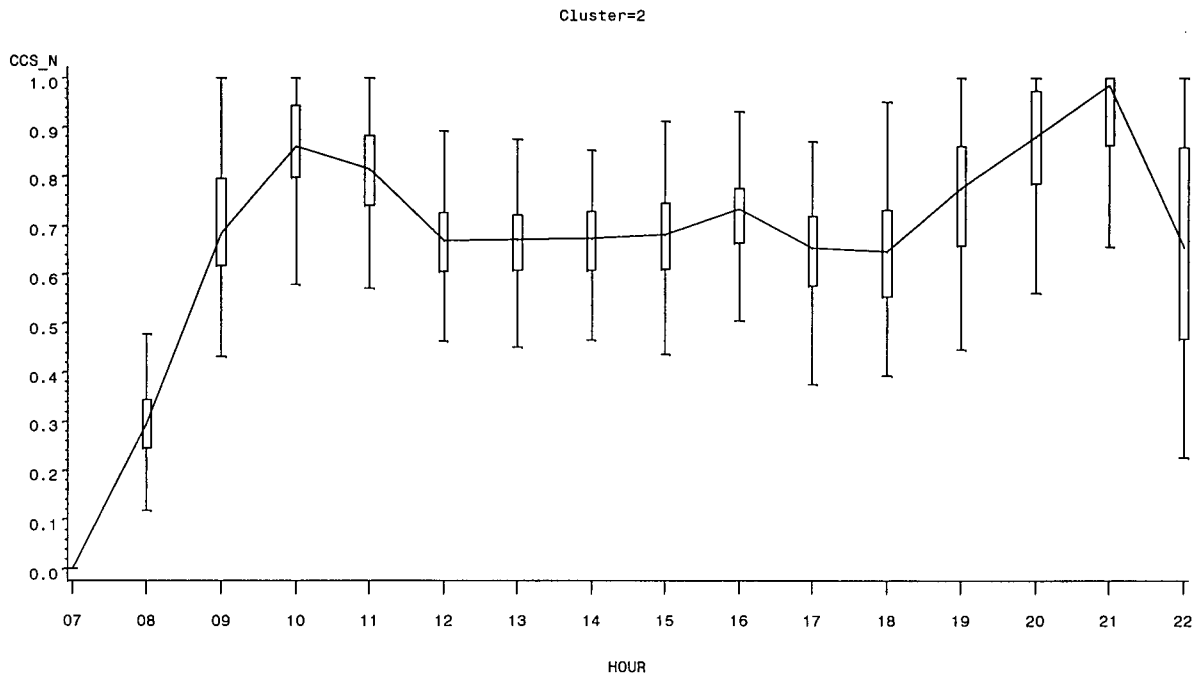
Appendix E Time series of the Outliers

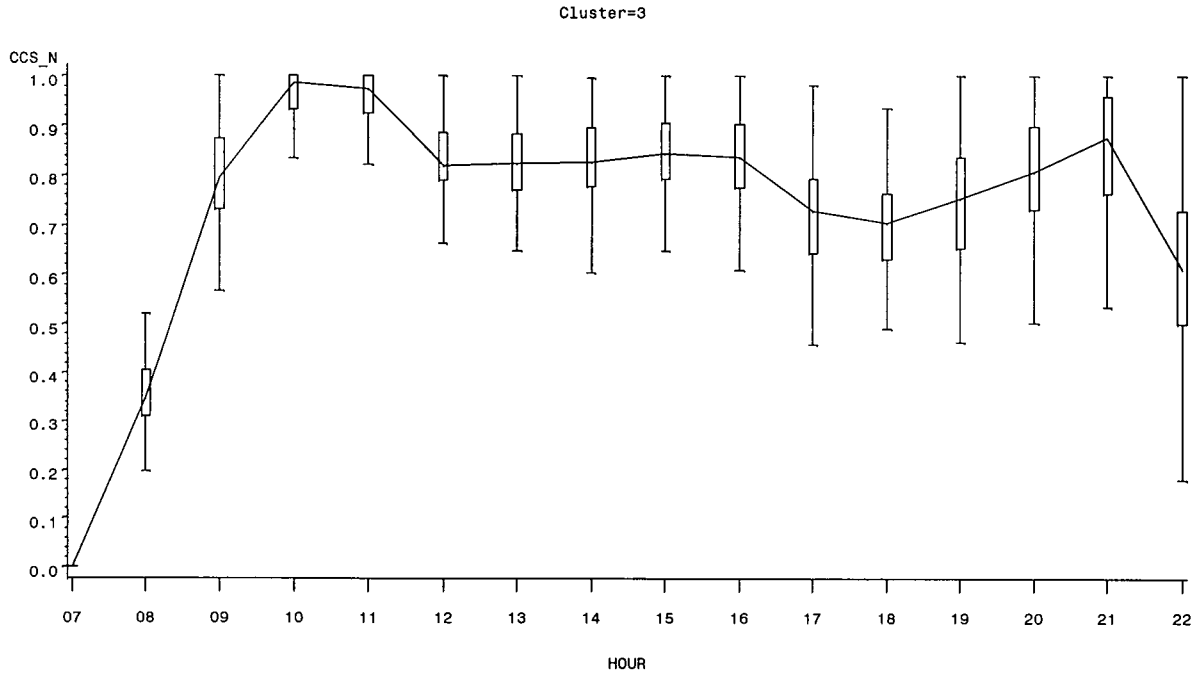


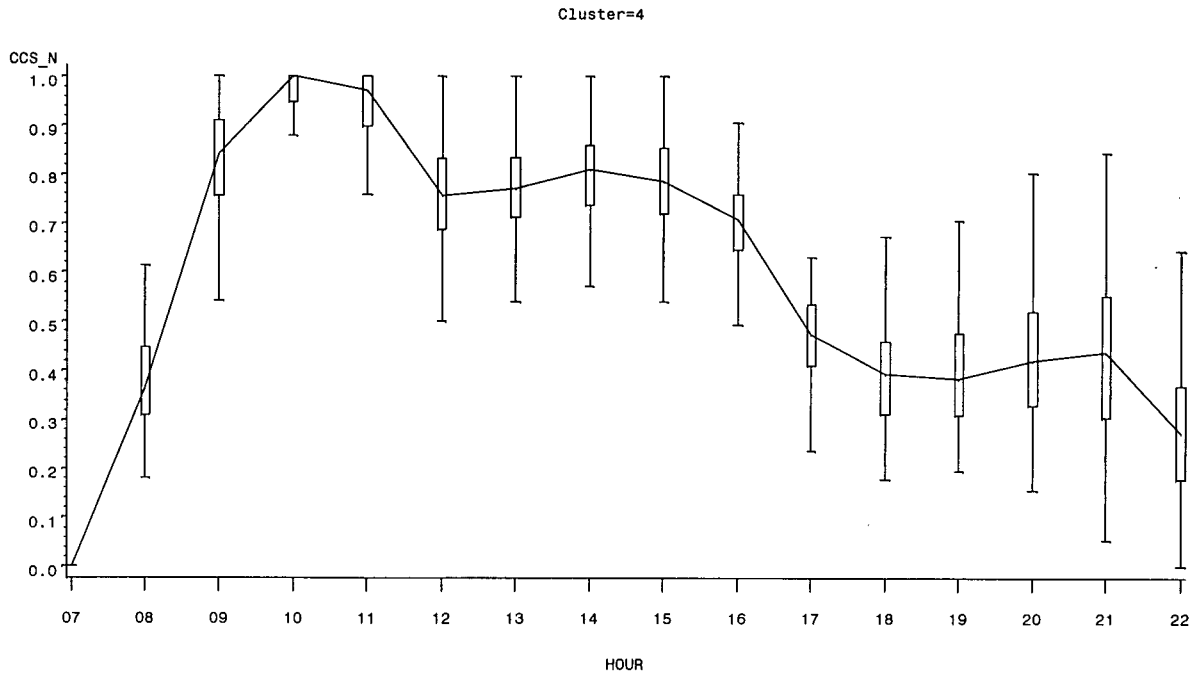


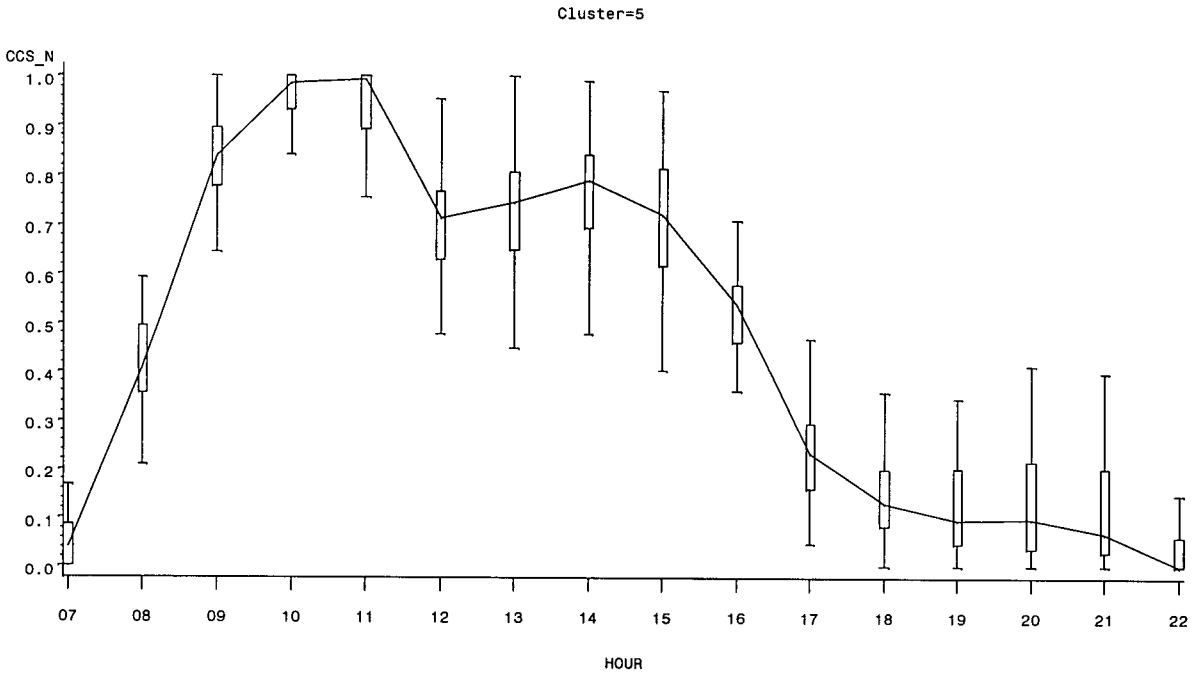
Appendix F Clusters obtained from using the FASTCLUS procedure





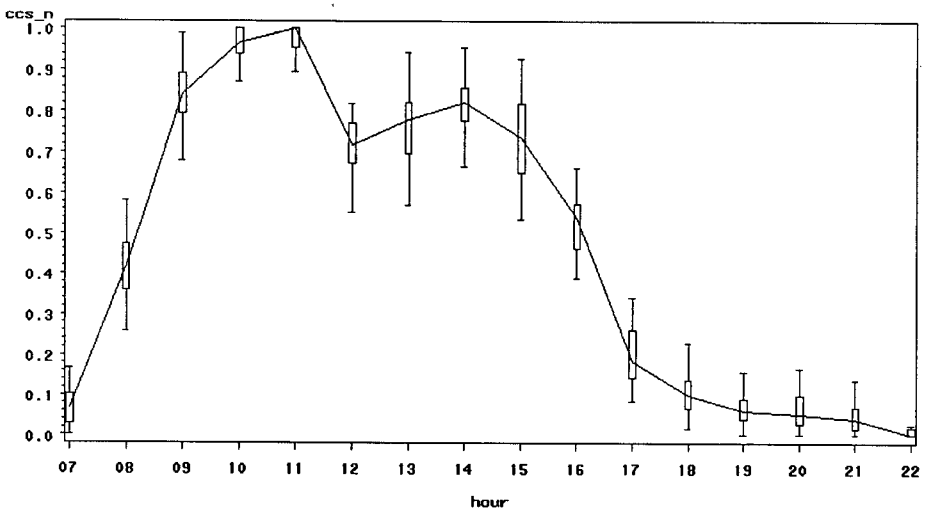

Cluster=3
CCS_N
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
HOUR


Cluster=4
CCS_N
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
HOUR

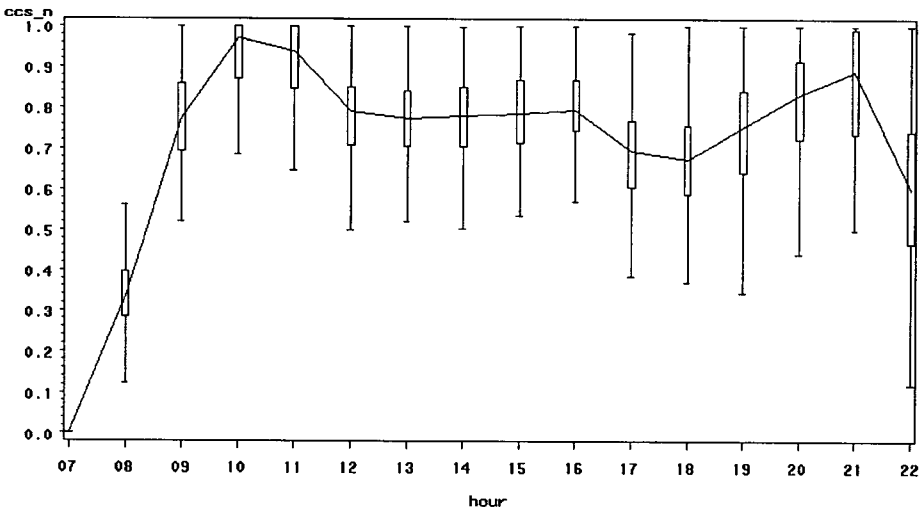

Cluster=5
CCS_N
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
HOUR

Appendix G Clusters obtained using the Average Linkage Method

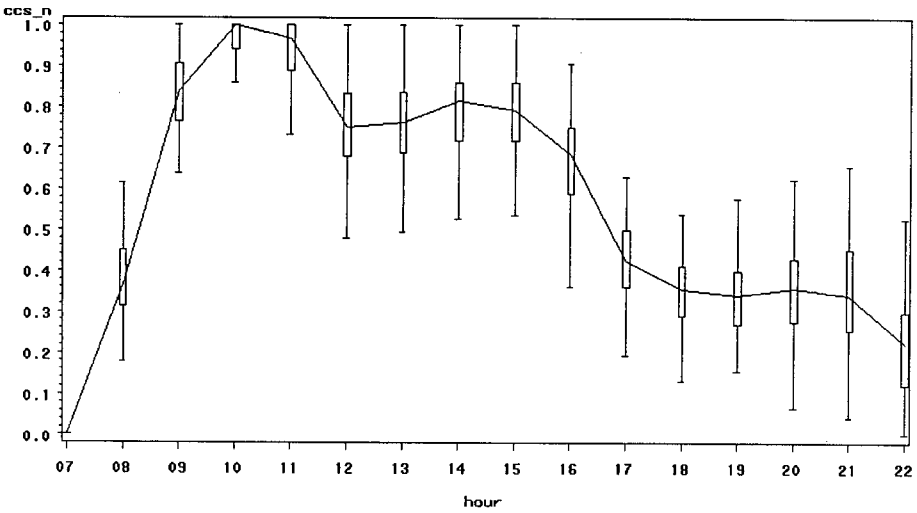
CLUSTER=1



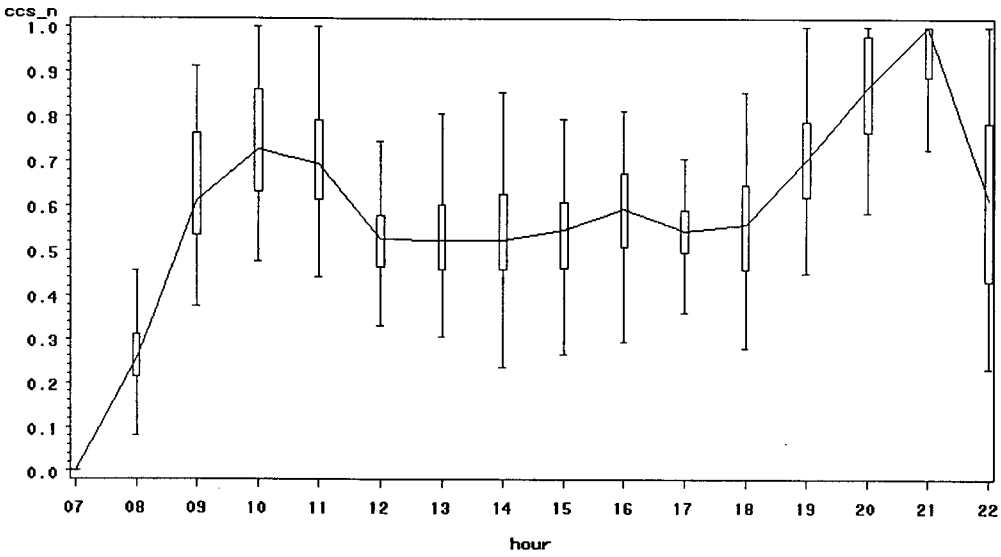
CLUSTER=2



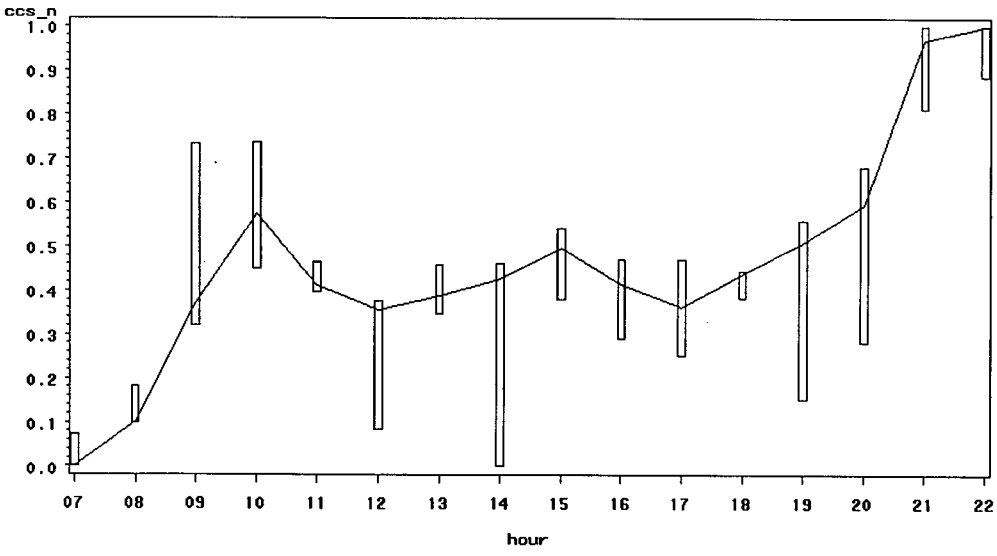
CLUSTER=3



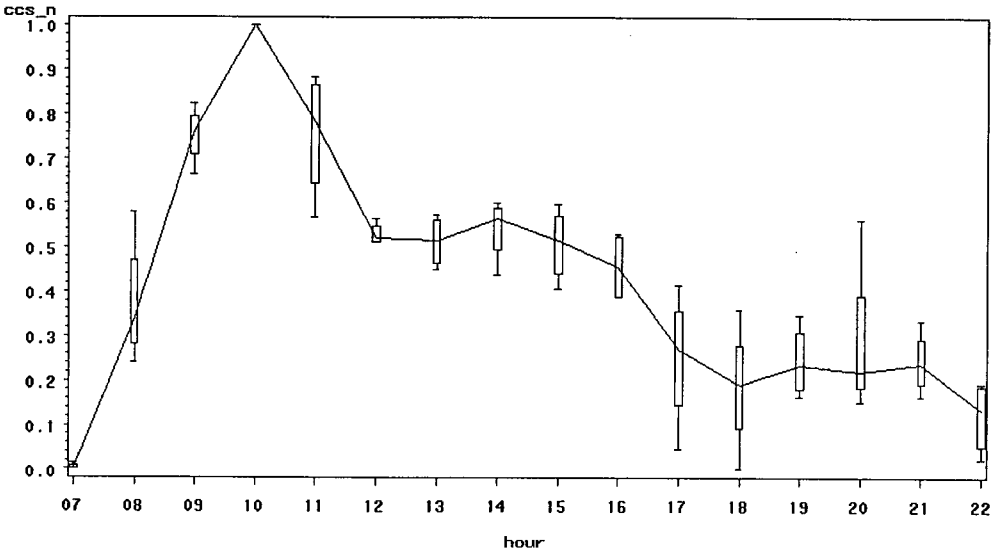
CLUSTER=4

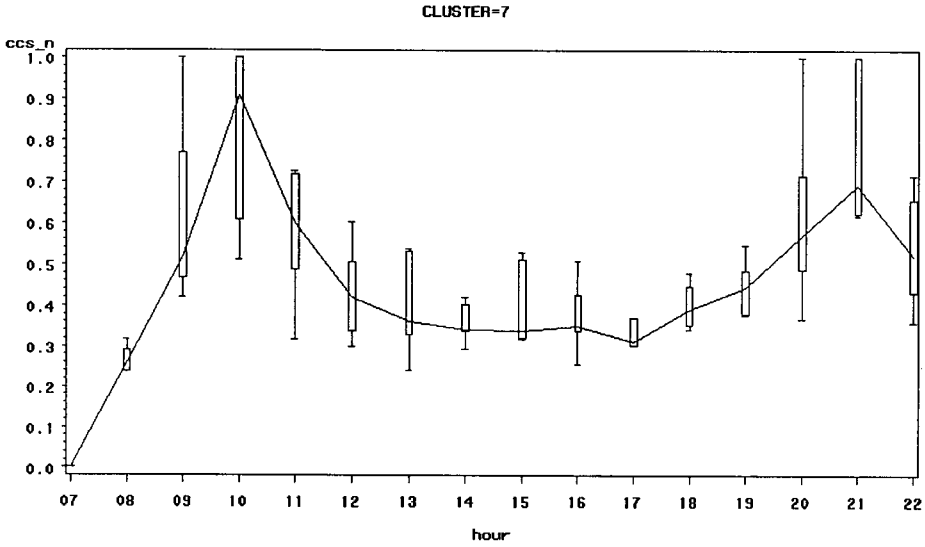


CLUSTER=5

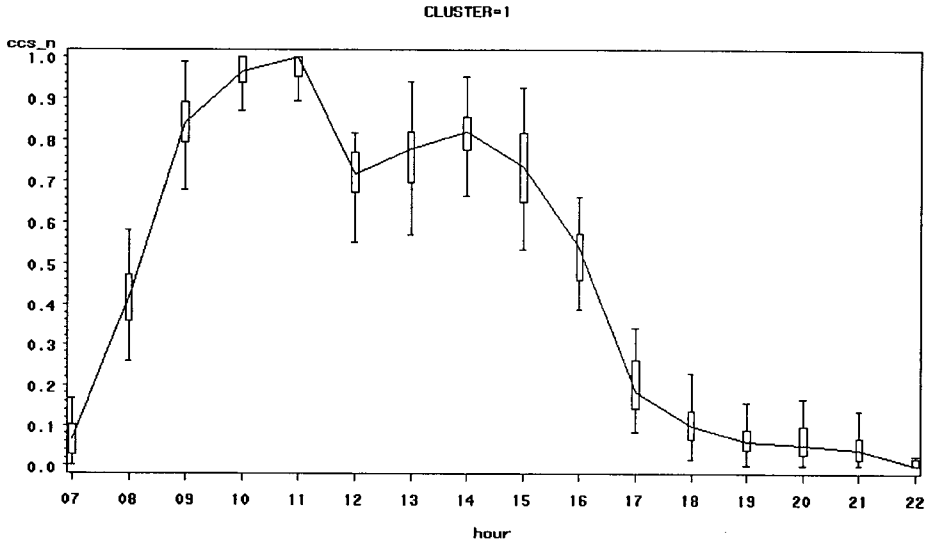


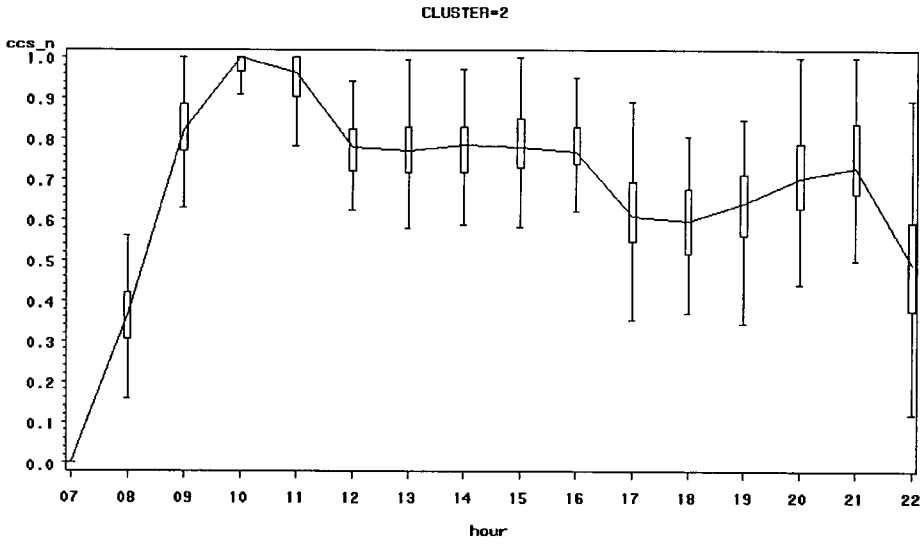
CLUSTER=6

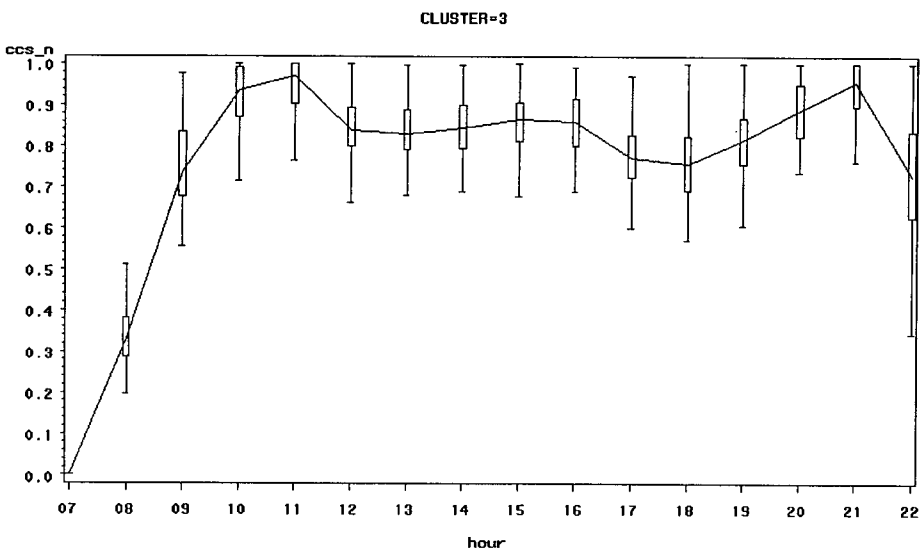


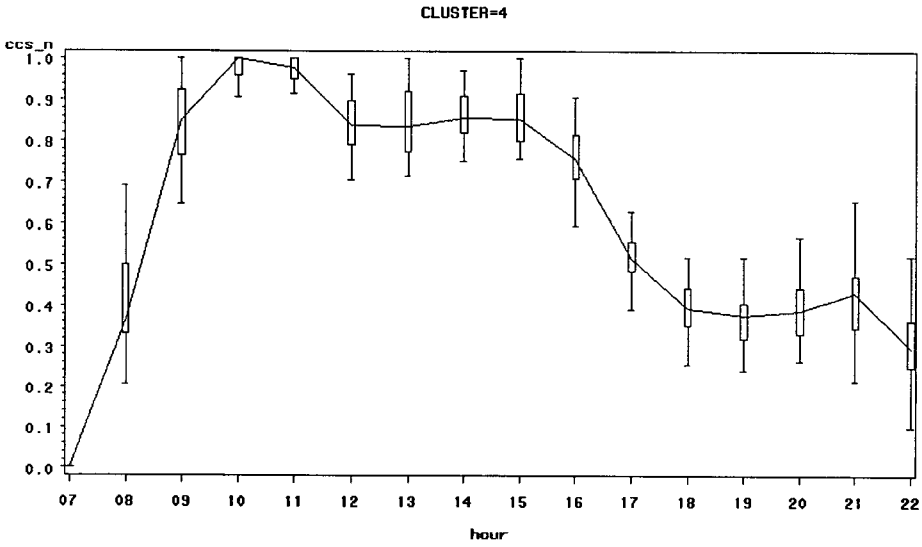

CLUSTER=7
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
hour

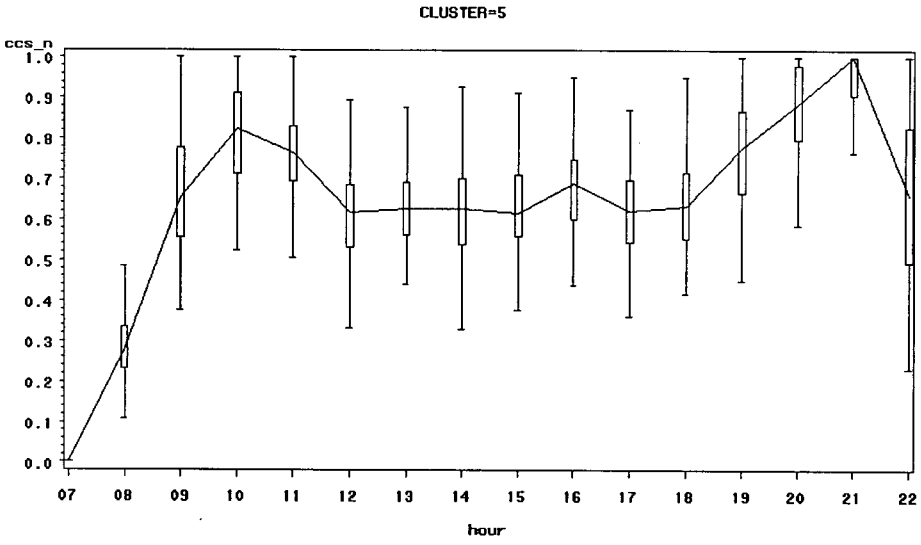
Appendix H Clusters obtained from using Ward's Method

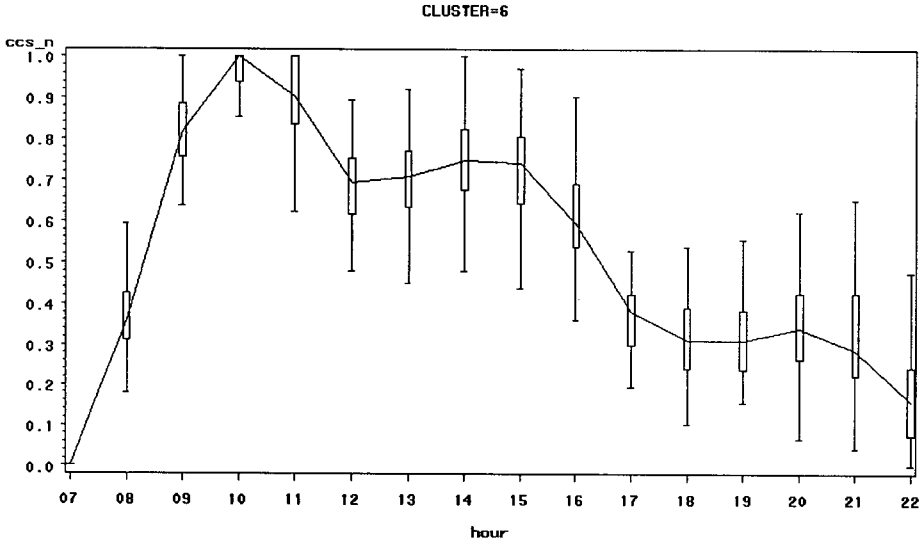


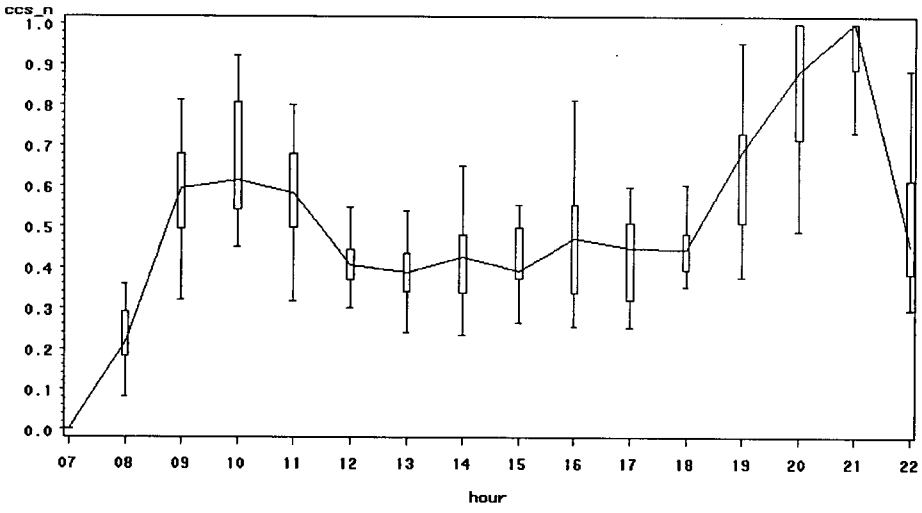




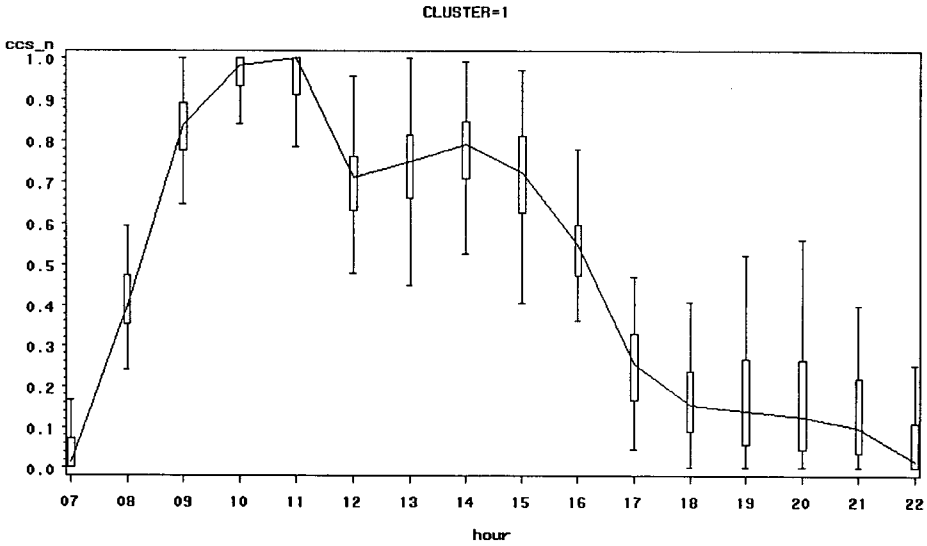

CLUSTER=4
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
hour

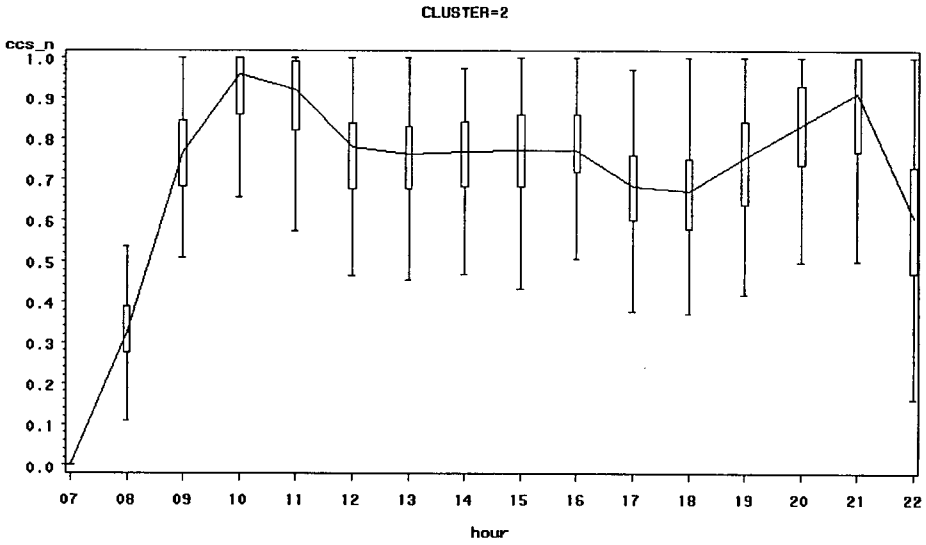

CLUSTER=5
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
hour

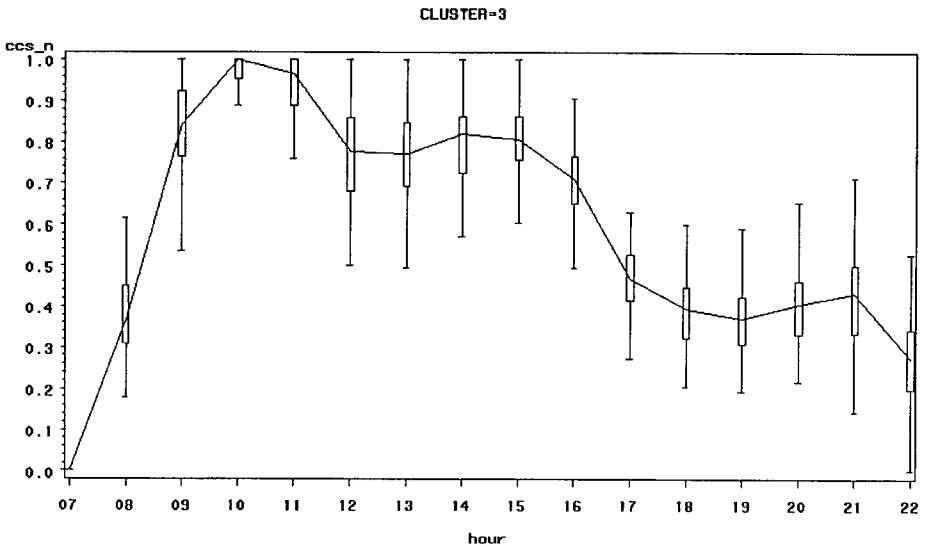

CLUSTER=6
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
hour

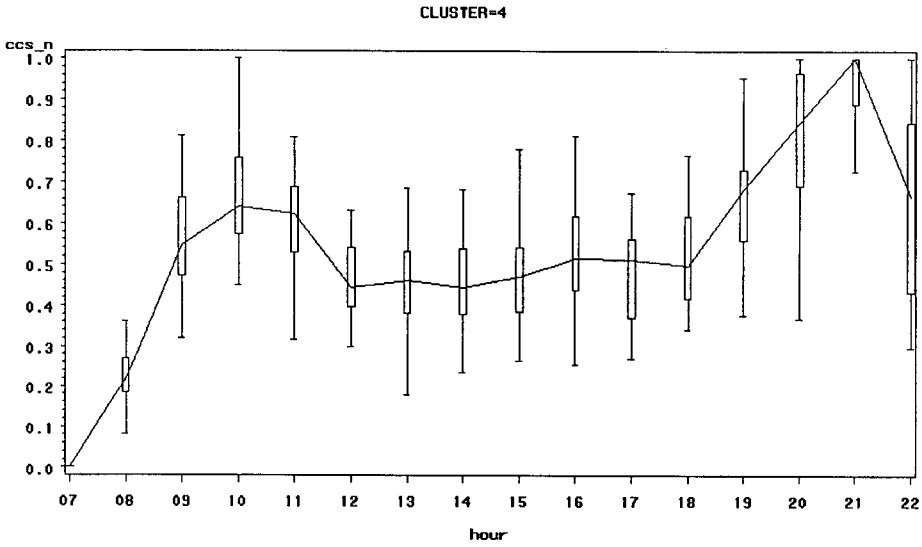

CLUSTER=7
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
hour

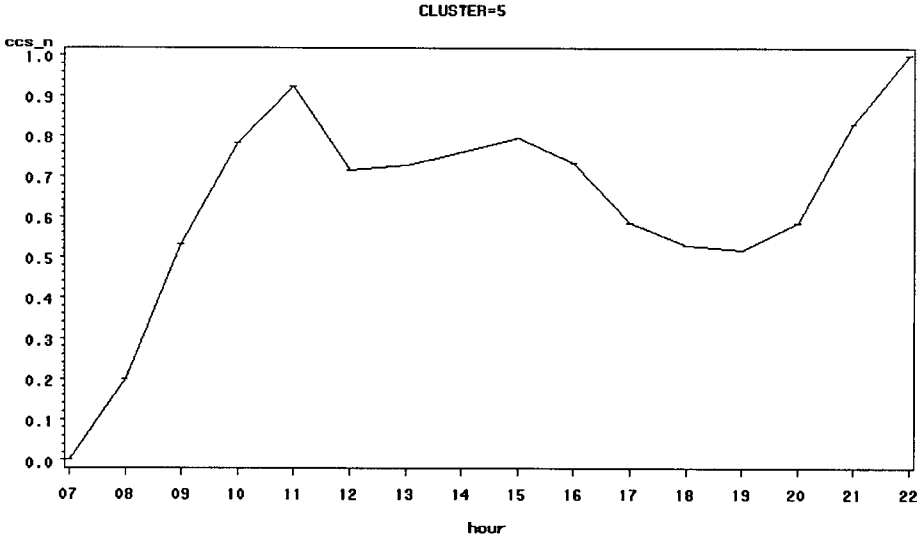
Appendix I Clusters obtained from using the Centroid Method

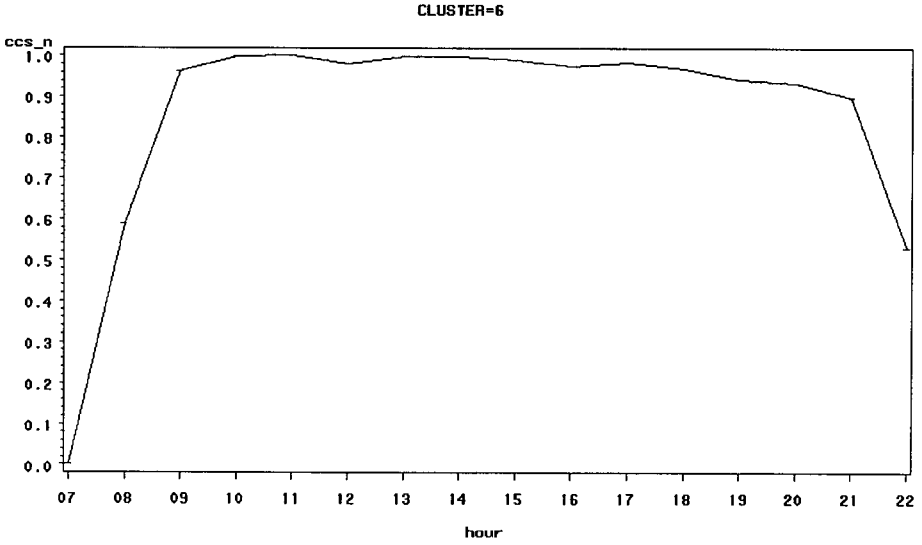







CLUSTER=4
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
hour


CLUSTER=5
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
hour


CLUSTER=6
ccs_n
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
hour

Appendix J Results of performing cluster analysis on raw data

The cluster analysis described in the results has been performed using the principal components as inputs. However, we have also tried using a variation of the methodology described in section 3. In this case, the cluster analysis was performed on the normalized raw data (instead of using the principal components obtained for the principal component analysis). The results obtained from this cluster analysis showed that the clusters obtained were similar to the ones obtained from performing the analysis using the principal components. This comparison was done by comparing the box-plots of the 7 clusters created by both analyses. This showed that the shape of the clusters were comparable. The final results using both methods were also compared. This showed that most of the new alternate routes obtained from both analyses were the same ones. However, even though the results seemed similar, we continued our analysis using the principal components as these gave more interpretable results. For example, the observations could be plotted in a three dimensional space and we could see visually if there were any peculiarities about the data or about how the clusters were formed.

Appendix K Results

AZ_id	tandem	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	H21	H22
01-02	16	Y	N	N	N	N	N	N	N	N	N	N	Y	N	Y	Y	Y
01-03	5	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	8	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	10	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	14	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	N	Y	Y
01-03	16	Y	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	N
01-03	20	Y	Y	N	N	N	N	N	N	N	N	N	N	N	Y	N	Y
01-03	23	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	24	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	25	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	26	Y	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	28	Y	N	N	N	N	N	N	N	N	N	Y	N	N	Y	Y	Y
01-03	30	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-03	31	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-05	0	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-07	0	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-08	0	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-09	0	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-10	0	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-10	16	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-10	28	Y	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y
01-11	16	Y	N	N	N	N	N	N	N	N	N	N	Y	N	N	Y	Y
01-11	26	Y	N	N	N	N	N	N	N	N	N	Y	N	Y	Y	Y	Y
01-12	0	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-13	5	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-13	7	Y	Y	N	N	N	N	N	N	N	N	Y	Y	N	Y	Y	N
01-13	8	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-13	10	Y	Y	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y
01-13	25	Y	N	N	N	N	N	N	N	N	N	Y	N	Y	N	Y	Y
01-13	27	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y
01-13	31	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-14	0	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
01-15	0	Y	Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
01-16	0	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y

Appendix L SAS code

```
*******************************************************************************
;
*       Filename:     Thesis - Application of Multivariate Anlysis to;
*                     Time of Day Routing;
*       Author:       Isabelle Smith;
*******************************************************************************
;
        options pagesize = 28 ls =100;
        options pagesize = 32 ls=120 nomprint;
        options obs=max;
*******************************************************************************
;
        %let st_hr=7;
        %let end_hr=22;

* The cutoff variable is used to determine the level at which an arc is busy or
not;
* This value is arbitrarily chosen;
        %let cutoff=0.7;
        *%let path1 =p:\telus\analysis\mva\analysis;
        %let path1 =C:\WINDOWS\Desktop\courses\TELUS\analysis;
        libname swork "&path1.\SASWORK";

*******************************************************************************
;
* Macro definitions;
*******************************************************************************
;
* Macro used to print;
        %macro pr(n);
               proc print data= &n; run;
        %mend pr;

* Macro used to append two nodes;
        %macro PAIR_ID (VAR1= ,VAR2=, CBN_VAR=);
                    length &CBN_VAR. $5.;
                if &var1 lt 10 then aa = compress('0' || put (&var1, 2.));
                else aa = put (&var1, 2.);
                if &var2 lt 10 then zz = compress('0' || put (&var2, 2.));
                else zz = put (&var2, 2.);
                if &VAR1 < &VAR2 then
                do;
                &CBN_VAR. =compress( PUT(aa, 2.) || "-" || PUT(zz, 2.));
                end;
                else do;
                &CBN_VAR. =compress( PUT(zz, 2.) || "-" || PUT(aa, 2.));
                end;
        %mend PAIR_ID;
```

```
*************************************************************************
;
* Read in the file containing AZ pair, Hour and CCS;
*************************************************************************
;
* The following file contains data for the 31st of July, 2000;
        *filename fn1 'p:\telus\analysis\personal\isabelle\cs.txt';
        filename fn1
'C:\WINDOWS\Desktop\courses\TELUS\analysis\saswork\cs.txt';

             data swork.cs;
            infile fn1 delimiter=',';
            input az_id $ hour1 CS;
            if az_id = 'az_id' then delete;
        run;

        data swork.data_01(drop=hour1 cs);
            format a z 2.;
            length a z 8.0;
            set swork.cs;
            hour = hour1-24; *the input data had the hours numbered from 24-47;
            CCS = CS/100;    *the input data had call volume in call seconds;
            a = substr(left(az_id),1,2);
            z = substr(left(az_id),4,2);
            if &st_hr <= hour <= &end_hr;
        run;

*************************************************************************
;
* Plot the time series of the arc utilization for each AZ pair;
*************************************************************************
;
/*
        symbol1 i=join v=plus;
        proc gplot data=swork.data_01;
                plot ccs *hour;
                by az_id;
        run; quit;
*/
*************************************************************************
;
* Normalize the CCS for each AZ pair to obtain CCS_n between 0 and 1
;
*************************************************************************
;

        * Find max and min ccs;
        proc summary data=swork.data_01 nway missing;
                class az_id;
                var ccs;
                output out=data_01b max=max_ccs min=min_ccs;
```

```
        run;

        proc sort data=swork.data_01; by az_id; run;
        proc sort data=data_01b; by az_id; run;
        data data_01c;
                merge swork.data_01(in=have)
                        data_01b(in=want);
                by az_id;
                                if have;
        run;
                * remove the two outliers that were found: 01-04 and 11-27;
        data data_01d;
                set data_01c;
            where az_id <> '01-04' and az_id <> '11-27';
        run;

        * Normalize CCS;
        data norm_01;
                set data_01d;
                ccs_n = (ccs - min_ccs) / (max_ccs - min_ccs);
        run;
        *%pr(norm_01);
       /*
       * Plot the time series of the normalized arc utilization for each AZ
pair;
        proc gplot data=norm_01;
                plot ccs_n * hour;
                by az_id;
        run; quit;
       */
*******************************************************************************
;
*       Prepare data for Principal Components Analysis;
*******************************************************************************
;
        proc transpose data=norm_01 out=Tnorm_01 prefix=N;
                by az_id a z;
                var ccs_n;
                id hour;
        run;


*******************************************************************************
;
*       Principal Components Analysis;
*******************************************************************************
;
        proc princomp data=Tnorm_01 out=swork.pca_out outstat=swork.pca_stat;
                var n&st_hr--n&end_hr;
        run;
/* scatterplots of the principal components
```

```
        symbol1 v=dot i=none h=0.5;
        proc gplot data=swork.pca_out;
                plot prin1 * prin2;
        run; quit;
        symbol1 v=dot i=none h=0.5;
        proc gplot data=swork.pca_out;
                plot prin2 * prin3;
        run; quit;
        symbol1 v=dot i=none h=0.5;
        proc gplot data=swork.pca_out;
                plot prin1 * prin3;
        run; quit;
*/
/*
        proc factor data=swork.pca_out nfactors=3 scree;
                var n&st_hr--n&end_hr;
        run;

        proc gchart data=swork.pca_out;
                vbar prin1;
        run; quit;

        proc sort data=swork.pca_out; by az_id; run;
        proc sort data=norm_01; by az_id; run;

        data test5;
                merge swork.pca_out(in=want keep=az_id prin1 prin2 prin3)
                    norm_01(in=have);
                by az_id;
            if have;
        run;

        %pr(test5);
        data test6;
                set test5;
            *where prin1<-4 and prin2<-2and prin3>0;
            where prin1<-4 and prin2<-2;
        run;
        symbol i=boxtj;
        proc gplot data=test6;
                plot ccs_n *hour/ overlay;
        run;quit;
*/
*************************************************************************
;
*       Cluster Analysis - Using fastclus procedure;
*************************************************************************
;
        proc fastclus maxclusters=5 maxiter=10 data=swork.pca_out
          out=swork.clus_01;
```

```
                var prin1 prin2 prin3;
        run;

/*
        symbol1 v=dot i=none h=0.5;
        proc gplot data=swork.clus_01;
                plot prin1 * prin2=cluster;
        run; quit;
        symbol1 v=dot i=none h=0.5;
        proc gplot data=swork.clus_01;
                plot prin2 * prin3=cluster;
        run; quit;
        symbol1 v=dot i=none h=0.5;
        proc gplot data=swork.clus_01;
                plot prin1 * prin3=cluster;
        run; quit;
*/
*******************************************************************************
;
*               Print CCS vs Shour for each cluster;
*******************************************************************************
;
        data data_pr1;
                set Tnorm_01(keep=az_id n&st_hr--n&end_hr);
        run;

        proc sort data=data_pr1; by az_id; run;
        proc transpose data=data_pr1 out=data_pr2;
                by az_id;
        run;

        data data_pr3;
                set data_pr2(rename=(col1=ccs_n));
                if (substr(left(_name_),2,2) < 10) then
                 hour=compress(0||substr(left(_name_),2,1));
                else hour=substr(left(_name_),2,2);
        run;

        proc sort data=data_pr3; by az_id; run;
        proc sort data=swork.clus_01; by az_id; run;
        data clus_02;
                merge data_pr3(in=have)
                swork.clus_01 (in=want keep=az_id cluster);
            by az_id;
            if have and want;
        run;

        proc sort data=clus_02; by cluster; run;
                symbol1 i=boxtj;
        proc gplot data=clus_02;
                by cluster;
```

```
            plot ccs_n *hour;
        run; quit;
*******************************************************************************
;
*       Cluster Analysis - Using cluster procedure;
*******************************************************************************
;
        %let numclus=7;

        proc cluster data=swork.pca_out method=ward pseudo ccc outtree=tree ;
                var prin1 prin2 prin3;
                id az_id;
        run;
/*
        symbol1 v=plus i=none;
        axis1 label=(angle=90);
        proc gplot data=tree;

        plot _ccc_ * _ncl_ /vaxis = axis1 haxis=0 to 16 by 2;
        run;quit;
*/
        proc tree data=tree out=treeout nclusters=&numclus;
                copy prin1 prin2 prin3;
                id az_id;
        run;

*******************************************************************************
;
* plot normalized time series by cluster;
*******************************************************************************
;
        proc sort data=data_pr3; by az_id; run;
        proc sort data=treeout; by az_id; run;
        data clus_03;
                merge data_pr3(in=want)
                treeout(in=have keep=az_id cluster);
            by az_id;
            if have;
        run;

        symbol1 i=boxtj;
                proc sort data=clus_03; by cluster ; run;
        proc gplot data=clus_03;
                by cluster;
            plot ccs_n *hour;
        run;quit;

*******************************************************************************
;
* find 2 adjacent complementary arcs;
```

```
*******************************************************************************
;
/*
%let gr1=1;
%let gr2=2;
*/
        %let outfile= outcl5;
        filename myfile "p:\telus\analysis\mva\analysis\saswork\&outfile";
             filename myfile
"C:\WINDOWS\Desktop\courses\TELUS\analysis\saswork\&outfile";

%macro makefile(gr1,gr2);
* Form two groups to be used to find adjacent pairs of arcs;

        proc sort data=clus_03; by az_id cluster; run;
        proc summary data=clus_03 nway missing;
                        by az_id cluster;
                        output out=clus_04;
        run;

        data group1(drop=az_id);
            format a z $2.;
            set clus_04(keep=az_id cluster);
            a = substr(left(az_id),1,2);
            z = substr(left(az_id),4,2);
            az_id1 = az_id;
            where cluster=&gr1;
        run;

        data group2(drop=az_id cluster);
            format a z $2.;
            set clus_04(keep=az_id cluster);
            a = substr(left(az_id),1,2);
            z = substr(left(az_id),4,2);
            az_id2 = az_id;
            where cluster=&gr2;
        run;

* Find arcs in group 2, for each AZ pair in group 1, that start with the same
"a" node;

        proc sort data=group1; by a; run;
        proc sort data=group2; by a; run;
        proc sql;
                create table start_a as
                select group1.a, group1.z, az_id1, az_id2a
                from group1, group2(keep=a az_id2 rename=(az_id2=az_id2a ))
                where group1.a=group2.a;
```

```
* Find arcs in group 2, for each AZ pair in group 1, that end with the same "a"
node;

        proc sort data=group1; by a; run;
        proc sort data=group2; by z; run;
        proc sql;
                create table end_a as
                select group1.a, group1.z, az_id1, az_id2a
                from group1, group2(keep=z az_id2 rename=(az_id2=az_id2a z=a))
                where group1.a=group2.a;


* Append the table containing arcs that start with "a" with the table
containing arcs that end with "a";

        data match_a;
                set start_a end_a;
        run;

        proc sort data=match_a; by a z; run;


* Find arcs in group 2, for each AZ pair in match_a, that start with the same
"z" node;
* Note: we only need to look at AZ pairs in match_a, not all of the AZ pairs in
group1;

        proc sort data=match_a; by z; run;
        proc sort data=group2; by a; run;
        proc sql;
                create table start_z as
                select match_a.a, match_a.z, az_id1, az_id2a, az_id2z
                from match_a, group2(keep=a az_id2 rename=(az_id2=az_id2z a=z))
                where match_a.z=group2.z;


* Find arcs in group 2, for each AZ pair in match_a, that end with the same "z"
node;

        proc sort data=match_a; by z; run;
        proc sort data=group2; by z; run;
        proc sql;
                create table end_z as
                select match_a.a, match_a.z, az_id1, az_id2a, az_id2z
                from match_a, group2(keep=z az_id2 rename=(az_id2=az_id2z))
                where match_a.z=group2.z;


* Append the table containing arcs that start with "z" with the table
containing arcs that end with "z";
```

```
        data match_z;
                set start_z end_z;
        run;

        proc sort data=match_z; by a z; run;


* Keep the observations for which the az_id2a and the az_id2z are adjacent;

        data match(drop=M);
                set match_z;
                M=0;
                if (substr(left(az_id2a),1,2) = substr(left(az_id2z),1,2)
                or substr(left(az_id2a),1,2) = substr(left(az_id2z),4,2)
                or substr(left(az_id2a),4,2) = substr(left(az_id2z),1,2)
                or substr(left(az_id2a),4,2) = substr(left(az_id2z),4,2))
                then M=1;
                          if M=1;
        run;


*************************************************************************
;
* find if the available capacity on the adjacent arcs is sufficient;
*************************************************************************
;
             data avail_01;
                    set norm_01;
                    avl_ccs = max_ccs - ccs; * use capacity instead of
max_ccs...;
             run;

* merge available capacity and max_ccs on AZ pair;
             proc sort data=avail_01; by az_id hour;
             proc sort data=match; by az_id1;
             proc sql;
                create table match_01 as
                select hour, match.a, match.z, match.az_id, avl_ccs, max_ccs,
az_id2a, az_id2z
                from match(rename=(az_id1=az_id)), avail_01(keep=avl_ccs
max_ccs az_id hour)
                where match.az_id=avail_01.az_id;


* merge available capacity and max_ccs on first arc of alternate route;
             proc sort data=match_01; by az_id2a hour;
             proc sql;
                create table match_02 as
                select match_01.hour, match_01.a, match_01.z, match_01.az_id,
          avl_ccs, max_ccs, match_01.az_id2a, avl_ccsa,max_ccsa, az_id2z
```

```
            from match_01, avail_01(keep=avl_ccs az_id hour max_ccs
        rename=(max_ccs=max_ccsa avl_ccs=avl_ccsa az_id=az_id2a))
            where match_01.az_id2a=avail_01.az_id2a and
             match_01.hour=avail_01.hour;


* merge available capacity on second arc of alternate route;
          proc sort data=match_02; by az_id2z hour;
          proc sql;
             create table match_03 as
             select match_02.hour, match_02.a, match_02.z, match_02.az_id,
        avl_ccs, max_ccs, match_02.az_id2a, avl_ccsa, max_ccsa,
        match_02.az_id2z, avl_ccsz, max_ccsz
            from match_02, avail_01(keep=max_ccs avl_ccs az_id hour
        rename=(max_ccs=max_ccsz avl_ccs=avl_ccsz az_id=az_id2z))
            where match_02.az_id2z=avail_01.az_id2z and
             match_02.hour=avail_01.hour;


* available capacity on alternate route is equal to the minimum available
capacity between;
* the first arc and the second arc: min(avl_ccsa, avl_ccsz);
* the High demand flag indicates "Y" when the AZ pair is using more than 70% of
its maximum arc utililization of that day;
* the Sufficient capacity flag indicates "Y" when the two arcs forming the new
route are using less than 70% of its maximum;
* arc utililization of that day and the available capacity on that route is
sufficient;

          data match_04(drop=a z az_id2a az_id2z);
                set match_03;
                avl_ccsr = min(avl_ccsa, avl_ccsz);
                High_dem = "N";
                Suff_cap = "N";
                if ((substr(left(az_id2a),1,2)=substr(left(az_id2z),1,2))
or              (substr(left(az_id2a),1,2) = substr(left(az_id2z),4,2)))
                then tandem=substr(left(az_id2a),1,2);
                if ((substr(left(az_id2a),4,2)=substr(left(az_id2z),1,2))
or              (substr(left(az_id2a),4,2) = substr(left(az_id2z),4,2)))
                then tandem=substr(left(az_id2a),4,2);
               if avl_ccs <0.3*max_ccs then High_dem="Y";
                if (avl_ccsa>max_ccsa*0.3)and (avl_ccsz>max_ccsz*0.3) and
                (avl_ccsr>max_ccs*0.05) then Suff_cap="Y";
                *if High_dem = "Y" and Suff_cap = "Y";
          run;


          proc sort data=match_04; by az_id tandem;
          proc transpose data=match_04 out=swork.list&gr1&gr2 prefix=H;
                by az_id tandem;
         var Suff_cap;
```

```
            id hour;

             data _null_;
                   set swork.list&gr1&gr2;
             file myfile mod;
             tmp1 = compress(AZ_id||","||tandem||",");
             put (tmp1) (9.) (H&st_hr--H&end_hr.) (3.0 ',');
        run;

%mend makefile;


%macro mkfiles;
        %macro header;
                %do i=&st_hr %to &end_hr;
                        %if &i=&st_hr %then %do;
                                "H&i"
                        %end;
                        %else %do;
                                ",H&i"
                        %end;
                %end;
        %mend;
        data _null_;
                file myfile;
        put 'AZ_id,tandem,'%header;
run;

        %do i=1 %to &numclus;
                %do j=1 %to &numclus;
                        %if (&i ne &j) %then %do;
                         %makefile(&i,&j);

                        %end;
                %end;
        %end;
%mend mkfiles; %mkfiles;

*******************************************************************************
;
* Determine how many hours to keep;
*******************************************************************************
;
        /*
* This section was used to determine during which periods were the AZ pairs;
* using more than a certain percentage (cutoff point) of their capacity;

        proc summary data=swork.data_01 nway missing;
                class az_id;
                var ccs;
                output out= test1 max=max_ccs min=min_ccs;
```

```
run;

proc sort data=swork.data_01; by az_id; run;
proc sort data=test1; by az_id; run;
data test2;
        merge swork.data_01 (in=have) test1(in=want keep=az_id min_ccs
     max_ccs);
        by az_id;
        if have;
run;

data test3;
        set test2;
        flag = 1;
        if ccs< (&cutoff * max_ccs) then flag=0;
        if flag=1;
run;

proc print data=test3; where hour<8; run;

* Findings: We can find that generally, AZ pairs are busiest between 8:00AM and
23:00PM;
        */
```