DEVELOPING PAVEMENT MANAGEMENT SYSTEMS FOR SMALL TO MEDIUM SIZE CITIES: CHALLENGES, SOLUTIONS, AND A CASE STUDY

A Thesis

by

SINA ABBASGHOLIPOUR

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee:	Nasir G. Gharaibeh
Committee Members:	Galen D. Newman
	Luca Quadrifoglio
Head of Department:	Robin Autenrieth

December 2018

Major Subject: Civil Engineering

ABSTRACT

In this research, three problems associated with developing a Pavement Management System (PMS) for small to mid-size cities are discussed, and a solution is suggested for each of these problems.

First, the comparison of condition of the road network based on PCI and IRI indicates that using IRI as a pavement condition indicator in urban areas can lead to misleading information about the condition of the road network. Additionally, the relationship between IRI and PCI was assessed. It was demonstrated that there is a weak correlation between IRI and PCI which is not enough to estimate one from another accurately. The lack of a meaningful relationship between these two pavement performance indicators in urban areas can be explained by distortions of IRI measurements in local road networks.

Next, the problem with developing pavement prediction models without having a historical condition database is raised and a solution is suggested to overcome this problem. It is shown that, by having a rough estimate of the overall change in condition of the road network during one year, an iterative process can be used to estimate pavement prediction models coefficient. The methodology is tested using data from the city of College Station and one prediction model was developed for each M&R treatment.

Finally, problems associated with segmentation of the road network in urban areas are discussed in more detail. An automated segmentation method based on PDA approach is developed in Python. The segmentation code is compatible with ArcGIS which provides the user with all the visualization and analytical tools in ArcGIS. This segmentation method is successfully implemented for the city of College Station’s road network considering four different thresholds.

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to Professor Nasir Gharaibeh for his continuous support of my research, for his patience, and his trust in me. This thesis would not be possible without his guidance and immense knowledge.

I also would like to express my gratitude to members of my committee, Professor Luca Quadrifoglio and Professor Galen Newman for their patience and support.

I also would like to thank my parents and my sister whom without I would never be the person I am today. For that and their unfailing support I wish to express my deepest gratitude.

My heartfelt thanks go to Kiana whose untiring help and support have always been with me when I needed them the most.

I thank all my friends in particular Hooman and Nathan for always being there for me and making my time at Texas A&M a wonderful experience.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a dissertation committee consisting of Professor Nasir Gharaibeh and Professor Luca Quadrifoglio of the Zachery Department of Civil Engineering, and Professor Galen Newman of Department of Landscape Architecture and Urban Planning. All work for the dissertation was completed independently by the student.

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

TABLE OF CONTENT

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1 INTRODUCTION

1.1 Background

Pavement Management Systems (PMSs) were first developed in the 1970s. Since then, they are considered as helpful tools for Departments of Transportation (DOTs) and other agencies to manage their maintenance and rehabilitation activities, and to allocate their budget in the most cost-effective way.

Developing a PMS consists of several steps. Firstly, the road network is surveyed for assessment of the road condition. After recording the pavement distresses, a condition index is generated based on the field data. Lastly, Life Cycle Cost Analysis (LCCA), performance prediction models, and benefit calculations are conducted for alternative treatment types. A satisfactory PMS result in a comprehensive Maintenance and Rehabilitation (M&R) plan which would reduce overall cost while maximizing the treatment benefit. Although PMSs have consistently been used to manage state and national highways for more than 40 years, the application of PMS to local urban areas is still limited. The common perception is that PMSs are not implemented by local agencies due to lack of resources. However, there is a lack of studies to understand the specific reasons for this problem and to develop PMS tools suitable for local agencies.

This study is motivated by two key questions:

1. What are the technical problems that limit the implementation of PMS, in local urban areas?
2. What are possible solutions to these problems?

1.2 Research Objectives

The aim of this research is to identify some of the common problems facing the development and implementation of PMSs for small to medium size towns and cities. Furthermore, appropriate solutions are suggested to remove these obstacles in order to facilitate the development of PMSs for urban areas. Based on a review of the literature, three technical obstacles to implement PMSs for small to medium size cities and towns have been identified, as follows:

- Misuse of IRI.
- Lack of a segmentation method.
- Lack of historical pavement condition data to calibrate performance prediction models.

The specific objectives of this research are:

1. To examine the suitability of International Roughness Index (IRI) for measuring the condition of urban roads and explore possible relationship between Pavement Condition Index (PCI) and IRI for different functional classifications of road.
2. To develop a method for calibration pavement performance prediction models for roadway networks with no historical condition data.
3. To develop a roadway network segmentation method in a GIS environment for urban areas.

1.3 Research Task and Thesis Organization

The thesis consists of seven main sections. The materials to be covered under each section are described as follows:

- **Section 1: Introduction**

This section will provide a rationale for the research topic, the problem to be studied throughout the research, and overview of data to be used in the analysis.

- **Section 2: Literature Review**

Review of existing literature on related topics is covered in this section. It contains discussions of key topics including, performance prediction models for the pavement networks, available PCI prediction models, use of IRI or other condition indices, network segmentation methods and advantages/disadvantages of each.

- **Section 3: Assessing the Suitability of Using IRI as a Pavement Condition Indicator for Urban**

This section analyzes data from the city of College Station to determine if IRI is a suitable measure of pavement condition and to determine if there is any correlation between IRI and PCI for various roadway classification. Furthermore, the correlation between IRI and PCI is evaluated using ANOVA (F-test) and other statistical tools.

- **Section 4: Development of a Method for Calibrating Pavement Performance Prediction Models for Networks without Historical Pavement Condition Data**

In this section of the research, a new method for calibrating pavement deterioration models is suggested which does not require any historical data. All the steps for creating the model are explained in details. This method is used to develop pavement performance prediction models for the city of College Station.

- **Section 5: Development of a Roadway Network Segmentation Method in ArcGIS**

This section describes the development of a roadway network segmentation method in ArcGIS. The program is developed in Python. Different scenarios of segmentation are performed and the results are compared.

- **Section 6: Summary of Research Efforts, Conclusions and Recommendations**

This section provides a comprehensive summary of the overall research efforts, the conclusions of the study, and recommendations for future studies.

2 LITERATURE REVIEW

2.1 Pavement Management System for small to mid-size cities

Attentions toward developing PMS methods for small to medium size cities have been increased to a great extent in the recent years. The increasing cost of treatments due to inflation, municipalities’ limited budgets which have been decreasing due to financial crisis, the emergence of heavier trucks which has led to more severe damages to the pavement, and the constantly growing number of trucks have all lead to the necessity of using a systematic method in order to allocate the annual budget in the most cost-effective way.

There are three main differences between managing the urban road network and the inter-urban road network. First, as some studies have suggested, the main serviceability index for highways is different from the urban road network. Roughness is usually considered to be the main serviceability index for highways, however, for the urban road network, pavement surface condition is usually the most crucial factor while developing a PMS. Second, pavement prediction models for urban pavements are more complicated than inter-urban pavements due to pavement interventions, utility cuts for installation of underground facilities, singularities such as manholes, and speed bumps. Third, segmentation of the road network in urban areas is more complicated than the inter-urban road network due to the existence of small sections in the road network. In addition, traffic flow and pavement material in the urban road network change more frequently from one section to another in comparison to inter-urban road network which leads to the creation of smaller segments (Osorio et al., 2014).

One of the very first efforts to establish a structure for developing PMS for small cities goes back to 1992. In this year, Tavakoli et al. proposed a procedure for creating Pavement Management System for small communities (PMSC). Based on this study, seven steps need to be taken for developing a PMSC. First, a database should be created including different information of roads such as section identification, pavement and shoulder characteristics, drainage system, traffic flow, and historical information. In the next step, type, area, and severity of distresses are surveyed and Pavement Condition Index (PCI) is calculated for each section of the road. Next, a group of alternative treatments are defined to give the user the freedom of choosing a treatment out of possible M&R strategies. Then, a unit cost and a deterioration model is assigned to each treatment option with regard to the extension of the M&R treatments. Subsequently, a Priority Index (PI) is computed for each strategy using Equation 2-1 (Tavakoli et al., 1992):

$$PI = \frac{1}{PCI} \times TF \times FC \times TR \times MF \qquad \textit{Equation 2-1}$$

Where PI = Priority Index; PCI = Pavement Condition Index; TF = Traffic Factor; FC = Functional Classification Factor; TR = Transit Route Factor; MF = Maintenance Factor (Tavakoli et al., 1992).

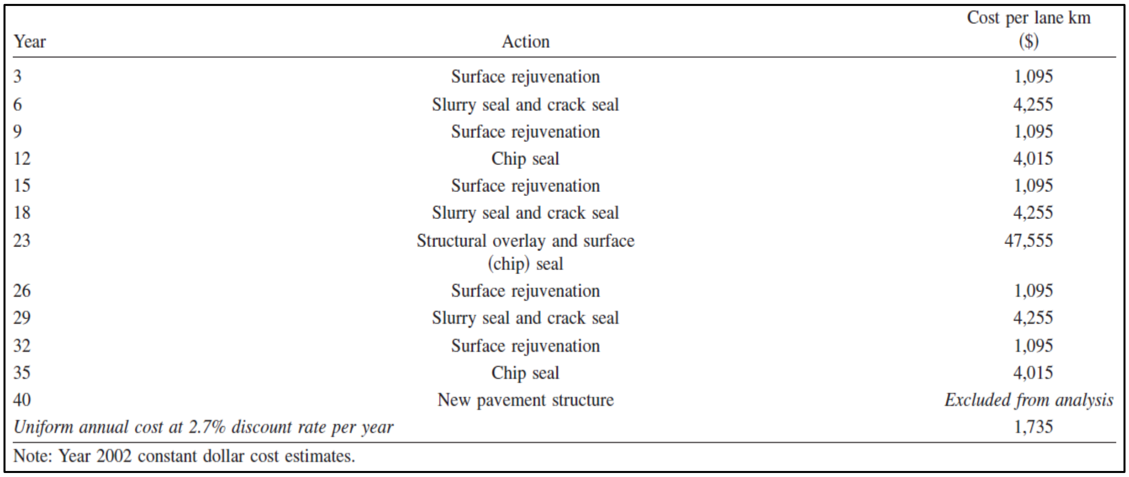
In the end, knowing PI and calculating the total maintenance and rehabilitation cost for all the sections, a strategy is assigned to each section of the road (Tavakoli et al., 1992).

Although this research was one of the pioneers in establishing a structure for developing a PMS for small cities, it is also suffering from oversimplification which can affect the accuracy of the PMS. In this study, the type of treatment in subsequent years is just a function of previous treatments, and the rate of deterioration and condition of the road is completely neglected. There is also no recommendations about the segmentation methods and how different sections are formed.

In 2009, Cottrell et al. described a method developed by Utah Department of Transportation (UDOT) for pavement maintenance management of small urban areas. In this method, M&R approaches are divided into three categories of preventive, corrective, and emergency. Preventive maintenance is usually used in early stages of formation of cracking and other types of distresses to reduce the rate of deterioration and postpone the implementation of corrective maintenance. Corrective maintenance, which is more expensive than preventive maintenance, is used for the repair of severe distresses. Emergency maintenance is usually used in the case of unpredicted problems and only provides short-term improvement (Cottrell et al., 2009).

Next, a fixed 40-year preventive maintenance is suggested for all low-volume roads with an average AADT of less than 5000, which can be seen in Table 2-1 (Cottrell et al., 2009). In this M&R plan, four different types of treatments are repeated periodically; and there is also a major rehabilitation after 23 years.

Table 2-1: Forty-year preventive maintenance cycle (Reprinted from Cottrell et al., 2009)



Year	Action	Cost per lane km ($)
3	Surface rejuvenation	1,095
6	Slurry seal and crack seal	4,255
9	Surface rejuvenation	1,095
12	Chip seal	4,015
15	Surface rejuvenation	1,095
18	Slurry seal and crack seal	4,255
23	Structural overlay and surface (chip) seal	47,555
26	Surface rejuvenation	1,095
29	Slurry seal and crack seal	4,255
32	Surface rejuvenation	1,095
35	Chip seal	4,015
40	New pavement structure	*Excluded from analysis*
Uniform annual cost at 2.7% discount rate per year		1,735

Note: Year 2002 constant dollar cost estimates.

The main problem associated with this method is that treatment is assigned to the road network without taking into account any information about the condition of the road and rate of deterioration.

In 2017, Giuseppe et al. proposed a group of strategies to ease the process of developing PMS in urban areas. In this method, first, PCI was calculated for the road network using ASTM

D6433. To take into account the effect of traffic in decision making, AADT was also calculated, and a comprehensive inventory including condition and traffic data was created using Geographical Information System (GIS). In this study, Vehicle Operating Cost (VOC) was used to calculate the benefit of each type of treatment. As it is shown below, VOC can be calculated using Equation 2-2. In this equation, VOC is a function of number of users, length and condition of a section of the road (Giuseppe et al., 2017).

$$VOC = \sum_{j=1}^{N} \left[365 \times AADT_j \times Length_j \times IRI_j \times \sum_{i=1}^{9} (Rate_i \times Price_i) \right] \qquad \textit{Equation 2-2}$$

VOC = annual Vehicle Operating Cost of the entire network; N = total number of sections; AADT = Annual Average Daily Traffic in section j; Rate = consumption rate of each component of VOC (tire wear cost, fuel consumption, repair and maintenance cost); Price = theoretical VOC of the main components of VOC when the IRI = 0; Length = length of section j; IRI = International Roughness Index of the section j; I = index of summation of the three VOC components per three different vehicles (nine in total); and j = index of summation of the total number of sections of the network (Giuseppe et al., 2017).

In order to calculate VOC, IRI should be estimated for each section of the road. To do so, IRI was calculated using Equation 2-3. In this equation, IRI can be estimated using PCI.

$$PCI = -0.224 \times (IRI) + 120.02 \qquad \textit{Equation 2-3}$$

A vital part of each PMS is to define a variety of strategies or treatment options and their respective performance prediction models. In this project, 5 alternative M&R strategies are defined to evaluate the effects of different funding policies. Change in PCI due to each of these strategies during a 5-year period can be seen in Figure 2-1 (Giuseppe et al., 2017).

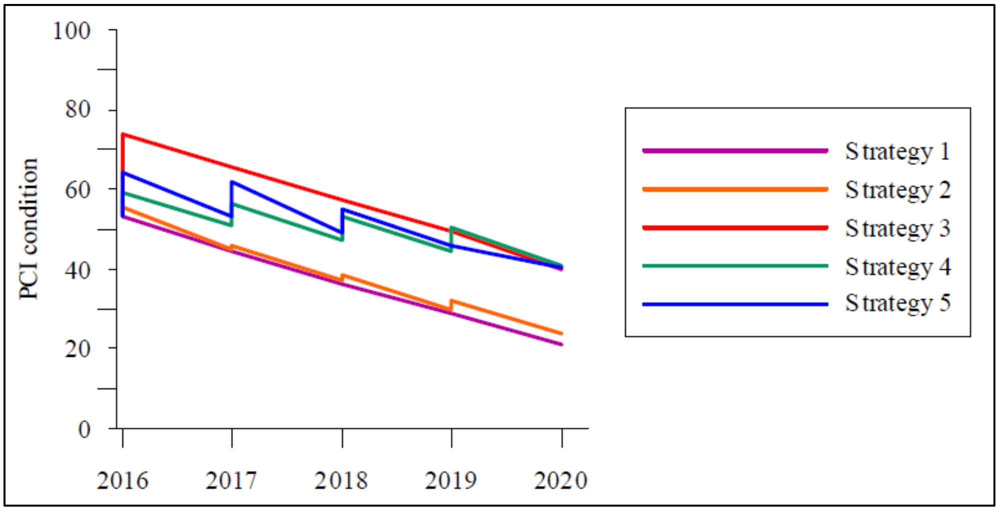


Figure 2-1: Performance prediction models for different strategies (Reprinted from Giuseppe et al., 2017)

In the last step, by calculating benefit (VOC) and cost (LCCA) of each strategy and comparing the results, different strategies were prioritized, and the best strategy was picked (Giuseppe et al., 2017).

Although the proposed method for developing PMS by Giuseppe is flexible and easy to implement in small to mid-size cities, it also has issues. For example, as it will be discussed in chapter 3 of this thesis, IRI cannot be estimated based on the PCI on a general basis for an urban area. Equation 2-3 was developed for District of Columbia in the USA and cannot be used for any other region. Another problem with this method is that a linear deterioration model was used to describe the change of PCI. However, pavement performance models follow more complicated models with deterioration rates that change with time.

2.2 Pavement Prediction Models

In 1986, Maria Nunez and Mohamed Shahin proposed a new method for forecasting the condition of the pavement in the future. The required data to perform analysis was provided by Paver Management System in the form of different pavement families for modeling and forecasting pavement condition. In this research, the concept of "pavement families" were first introduced; a group of sections which share similar characteristics such as pavement type, pavement use, and functional classification were known as a pavement family. In the next step, a screening process was conducted to eliminate possible errors. A statistical outlier analysis was performed to identify outliers which can have a negative effect on the accuracy of the prediction model. At the end, by using the screened database, a polynomial regression analysis was performed and different prediction models were developed regarding their pavement family. In this method, a prediction model was developed to predict the average behavior of each pavement family. For predicting the condition of a specific section, a parallel curve should be drawn to the original pavement family curve based on its specific condition and then, by using the curves, the PCI can be predicted in the future. An example of this procedure is presented in Figure 2-2 (Maria Nuez et al., 1986).

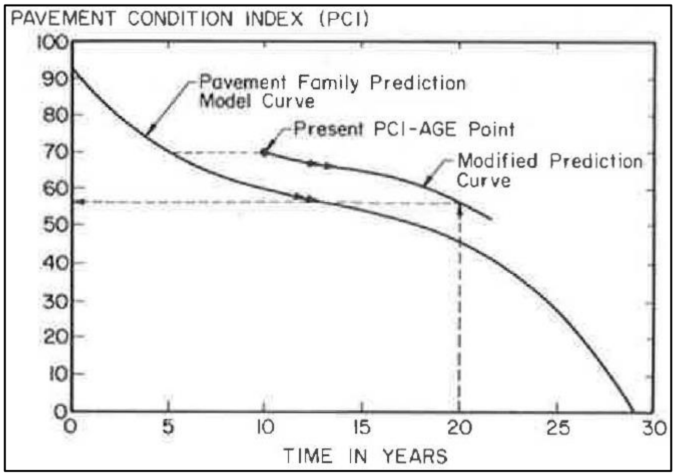


Figure 2-2: Example of pavement condition forecasting (Reprinted from Maria Nuez et al., 1986).

In 1989, K.P. George et al. concluded that mechanistic-empirical models associated with a historical database are the best fitted models for predicting deterioration of pavement. A preliminary analysis suggested different deterioration rates for each pavement family. Therefore, the data was divided into three classifications of flexible pavement with no overlay, flexible pavement with one or more overlays, and composite pavement. The summary of the regression for over 2000 miles of road network for each type of pavement is provided in Table 2-2. As can be seen, all models are suggesting a fairly strong coefficient of determination with relatively small standard error of estimate which suggest the developed Equations are quite satisfactory. The performance prediction model is presented in Equation 2-4.

$$\mathrm{PCR} = 90 - \mathrm{a}[\exp(\mathrm{Age}^{\mathrm{b}}) - 1]\log[\frac{\mathrm{ESAL}}{\mathrm{SNC}^{\mathrm{c}}}] \qquad \textit{Equation 2-4}$$

PCR: Pavement Condition Rating

SNC: Modified Structural Number

Age: Age since the last overlay, years

ESAL: Equivalent Single Axle Load

Table 2-2: Statistical parameters of PCR prediction models (Reprinted from George et al., 1989)

Type of Pavement	**Regression Coefficient**	**Standard Error of Estimate**	R^2
Flexible Original	a=0.6349	17.1	0.75
	b=0.4203	5.6	
	c=2.7062	21.4	
Flexible Overlay	a=0.8122	7.4	0.76
	b=0.3390	3.2	
	c=0.8082	38.8	
Composite	a=1.7661	3	0.69
	b=0.2826	6	

An example of developed models is also presented in Figure 2-3. It can be noticed that the rate of deterioration for flexible pavement with no overlay increases as the time passes. However, for the flexible pavement with overlay, the trend is almost linear and no obvious concave can be singled out.

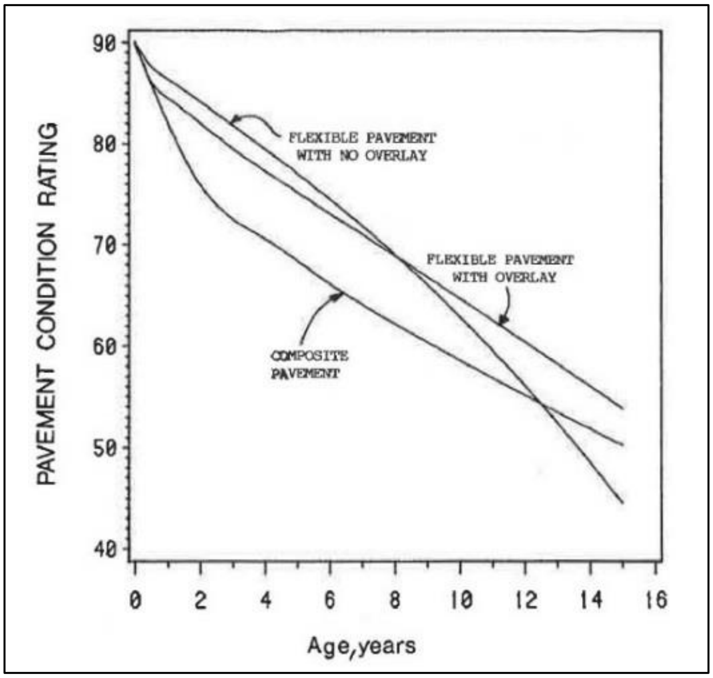


Figure 2-3: Pavement deterioration curves of three families of pavements (Reprinted from George et al., 1989)

In 2006, Rajagopal published a report in which pavement performance models were developed for the city of Cincinnati. Three variables were considered for developing the models which are as follows:

- Age: age of the pavement calculated as the number of years since the latest maintenance or rehabilitation.

- Functional classification of the roads: the roads are divided into subcategories regarding their functional classification. The categories include arterial freeways and highways, collectors, and local streets. These classifications also provide information about traffic flow and speed limits on each type of road.

- Pavement type and composition: at the beginning of developing a performance model, similar sections of the road—so-called homogenous sections—are put into a group to form practical construction sections.

Two different condition indices were used to develop the performance models. For major highways, Pavement Condition Rating (PCR) and for minor roads, Pavement Condition Number (PCN) was used to develop the models. A regression model was developed for each functional classification using the available historical data provided by the city of Cincinnati. The result of the regression model for each family is presented in Table 2-3. As can be seen, R^2 for major roads is between 0.4 and 0.6 which implies that the model can explain 40 to 60 percent of all the variability of response. However, R^2 for minor roads is only 0.2 which shows the model does not fit the data very well (Rajagopal, 2006).

Table 2-3: Summary of performance models (Reprinted from Rajagopal, 2006)

Family	**Performance index**	**Number of observations**	$\boldsymbol{R^2}$
Major Roads	PCR	91	0.51
Minor Roads	PCN	279	0.2

In 2015, Kan Wu proposed a PCI-based pavement management performance model for management of the road infrastructure system in fulfilment of his Master of Science thesis. In this research, PCI was automatically calculated by a programmed Excel sheet using LTPP pavement distress data. In this approach, a nonlinear optimization method was used to model a PCI vs. age master curve with parameters which differ for each pavement family. The result of this research displayed how historical PCI data can be used to develop a pavement performance master curve (Wu., 2015).

In 2015, a new integrated method for developing pavement performance prediction models was suggested by Guangyan Xu et al. In this research, which was conducted for Kentucky Transportation Cabinet (KYTC), historical data including 11 different condition attributes provided 4586 samples of the road network. For each individual distress score, a regression model was developed using available historical data to predict the distress score in the future. In the next step, a composite condition index was developed using all the indices. For calculating the composite condition index, an Analytic Hierarchy Process (AHP) analysis was conducted to calculate the weight of each of the 11 criteria. Similar to previous studies, the regression model was used to develop the performance prediction model, however, due to novel usage of regression models on each individual distress index and then calculating the composite index at the end, more accurate models were achieved with R^2 varying from 0.87 to 0.88. In Figure 2-4, the steps for creating deterioration prediction models are illustrated (Xu et al., 2015).

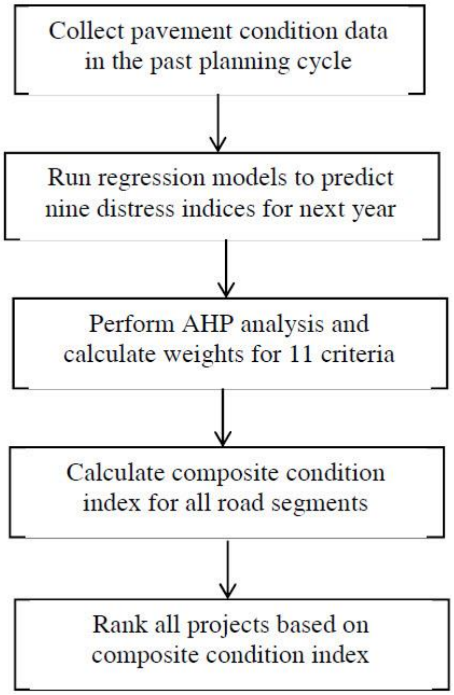

Collect pavement condition data in the past planning cycle
Run regression models to predict nine distress indices for next year
Perform AHP analysis and calculate weights for 11 criteria
Calculate composite condition index for all road segments
Rank all projects based on composite condition index

Figure 2-4: The integrated approach (Reprinted from Xu et al., 2015)

Various methods for developing performance models for road networks have been discussed. Although all the methods have essential differences in terms of the procedures used to develop the models, there is a common core in all the methods which is having historical data of the road network which enables researchers to conduct different types of analysis on data and predict the future condition. The problem with the previously mentioned methods is that without having historical data about the road network, none of these practices are applicable. Nowadays, small towns and municipalities are getting increasingly concerned with having a pavement management system of their own, however, most of these small towns do not own a historical database.

3 ASSESSING THE SUITABILITY OF USING IRI AS A PAVEMENT CONDITION INDICATOR FOR URBAN AREAS

3.1 Introduction

Evaluating the current condition of pavement is of great importance in order to develop cost-effective maintenance and rehabilitation plans.

Various pavement condition indices have been developed by different agencies to facilitate the management of road networks. These indices are used for:

- Developing Maintenance and Rehabilitation (M&R) plans
- Tracking changes in pavement condition
- Forecasting future pavement condition
- Budgeting

PCI and IRI are among the most commonly used indices in practice today. PCI was first developed in 1983 to evaluate pavement surface condition. To calculate PCI, a database including severity, quantity, and types of major distresses of pavement such as fatigue cracking, rutting, bleeding, depression, and potholes, etc. is created (ASTM, 2011). Then a deduct system is used where PCI is calculated from 0, being the worst possible condition, to 100, being the best possible condition. It is important to note that this method was originally developed to assess the pavement condition of roadway networks with relatively low traffic volume as well as airports, and parking lots.

On the other hand, IRI is the most frequently used index worldwide for expressing the smoothness of the pavement. This index is used by most of the highway agencies to assess the smoothness of the road by measuring the vertical movement of a quarter car simulation model over a specified distance (Gagarine, 2006).

The main objective of this study is to evaluate the suitability of using IRI as a pavement condition indicator of urban road networks. The study consists of three parts. First, the experimental PCI and IRI data from the City of College Station was used to compare network condition. In the second part, possible relationships between IRI and PCI in the local urban areas were investigated and correlation coefficient was measured for these two indices. In the last part, previously developed correlations between the two indicators were tested using the College Station data.

3.2 Literature Review

Although PCI and IRI were developed with different purposes and for different road networks, some researches have attempted to use these two indicators interchangeably. For example, Hajek et al. suggested that since Highway agencies are mostly using IRI to monitor the condition of pavement, local agencies should also adopt using IRI for their road networks. Hence, in this way, there would be a fair measure for all the agencies such as Highway agencies, and small municipalities to compare their road network condition, and be able to allocate their resources impartially (Hajek et al., 1998).

Additionally, there have been multiple studies to estimate PCI from IRI and vice versa. Dewan et al. (2002) conducted a study to investigate the correlation between IRI and PCI using the data from California Long-Term Pavement Performance (LTPP) sites (Dewan et al., 2002). In another study, in 2005, Dewan et al. established another power relation between PCI and IRI using 52 pairs of data collected on certain roads and streets in California. Since PCI and IRI data were not collected at the same time, PCI values were estimated using specific models for IRI time frame (Dewan et al., 2005). Another study which was conducted by Park et al. in 2007 suggested a parametric model for predicting PCI by using sample IRI from an available database (Park et al., 2007). The common characteristics of all these studies are that first, regression models were used to assess the relationship between PCI and IRI. Second, all the models suffer from having a low R-squared value, in one case as low as 18%, which indicates only a small variation of independent variable can be explained by dependent variable.

A study conducted in 2015 using a dataset of PCI and IRI samples collected over a span of 2 years in the District of Columbia suggests different regression models between IRI and PCI based on different functional classifications of road networks. Eight hundred and ninety five data points from different classifications of roads were used to develop these models. The Ordinary Least Squares (OLS) regression method was used to find the relationship between IRI and PCI. The results are shown in Table 3-1 (Arshin, 2015).

Table 3-1: Summary of regression analysis by functional classifications of roads (Reprinted from Arshin, 2015)

Functional Classification	**Model Equation**	**R^2**
Freeways	PCI = -0.215(IRI) + 110.73	0.56
Arterials	PCI = -0.206(IRI) + 114.15	0.71
Collectors	PCI = -0.217(IRI) + 115.32	0.73
Locals	PCI = -0.186(IRI) + 110.31	0.74

As can be seen in Table 3-1, R-squared values vary from 0.56 to 0.74 for different functional classifications of road networks which generally indicates an improvement compared to previous models. This improvement may be explained by dividing the network into functional classifications and assess each one of them separately rather than analyzing the whole network at once.

The main problem with these studies is that they neglect the initial assumptions associated with using IRI. The IRI originally was developed in response to a vehicle traveling with a constant speed of 50 mph on rural roads (Abudinen et al., 2017). However, due to stop and go traffic, and lower speed limit in comparison to rural roads, it is not possible to maintain such high constant speed in urban areas.

Furthermore, measuring IRI at a lower speed would increase the value of IRI by adding artificial roughness to the measured profile (Harrison et al., 2008). Additionally, IRI measurements can be affected by horizontal accelerations and decelerations of profiler when speed changes due to the traffic, and this behavior can even be magnified in lower speeds (Abudinen et al., 2017).

On the other hand, a pavement may seem smooth, but have serious distresses in reality (Al-Omari et al., 1995). Unlike IRI, PCI can be affected by low severity distresses such as low severity cracking like Alligator cracking, Block cracking, Joint cracking, and Longitudinal cracking. Additionally, there are distresses with no or little impact on IRI such as Bleeding, and Edge cracking. Although these types of distresses may not have any influence on the ride quality, they can lead to major distresses if not identified and maintained in early stages.

Last but not least, there are different challenges associated with measuring IRI in urban areas which are not often encountered in rural road networks. To name a few, the existence of speed bumps, manholes, drain inlets, and utility cuts can affect the IRI value. In addition, IRI measurements can be distorted due to railway crossings and rutted intersections. Although, encountering railway crossing and rutted intersections may happen infrequently, it can have a significant effect on IRI measurements, especially in short sections of a road network (Reggin et al, 2008).

In this study, we look at the suitability of using IRI in urban road networks with a new perspective; and with comparing the condition of network based on IRI with PCI using the College Station dataset.

3.3 Study Area and Data

Pavement management data from the City of College Station, Texas, was used to perform analysis. The data for 788 lane-miles of road network were collected in 2017 using automated methods. The functional classification of 3877 sections of the road network can be seen in Figure 3-1.

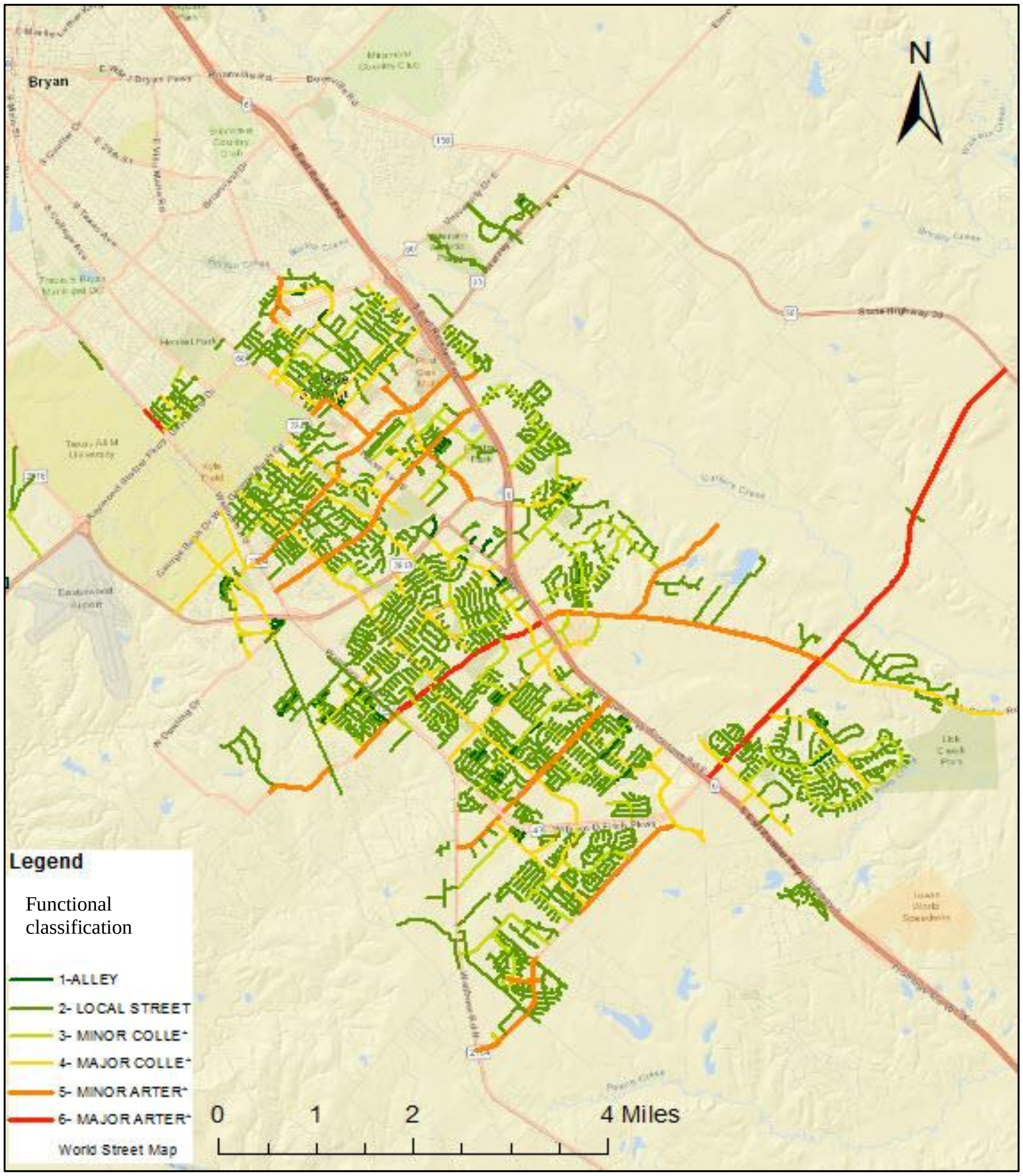


Figure 3-1: Map of road network functional classification of College Station, TX

To have a better understanding about variability in PCI and IRI, the distributions of PCI and IRI for the study area are shown in Figure 3-2.

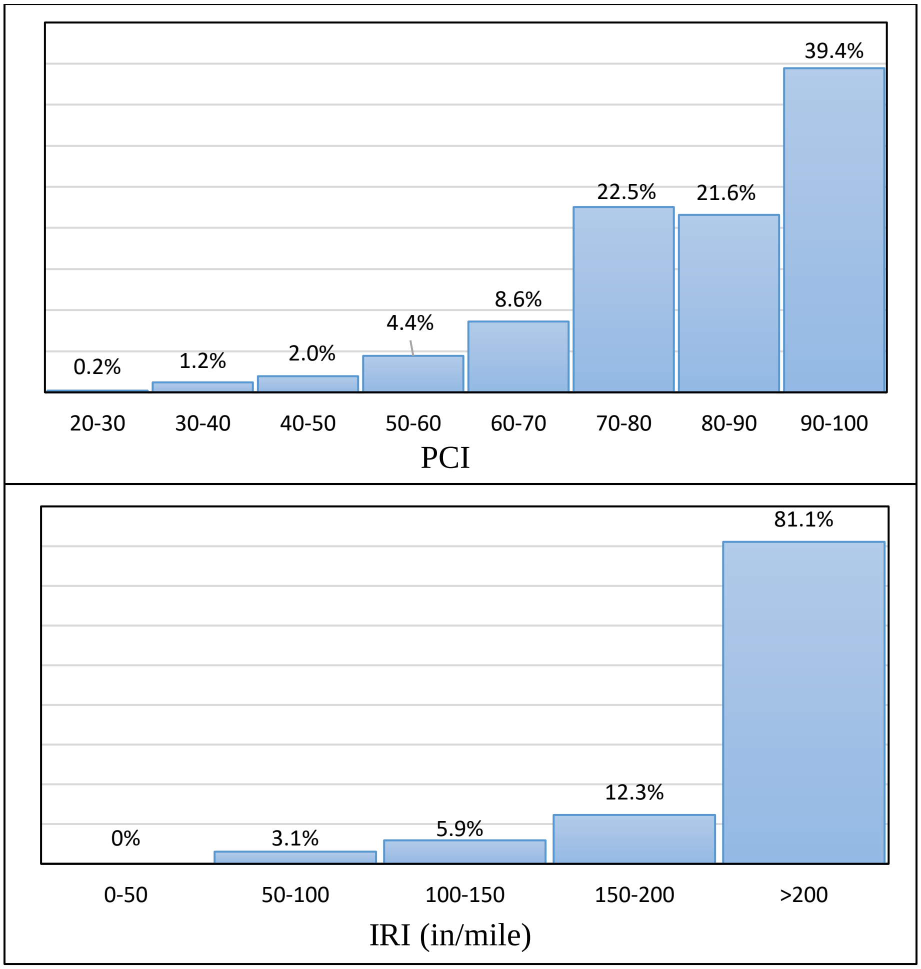


Figure 3-2: PCI frequency Distribution (Top), IRI frequency Distribution (Bottom) for College Station

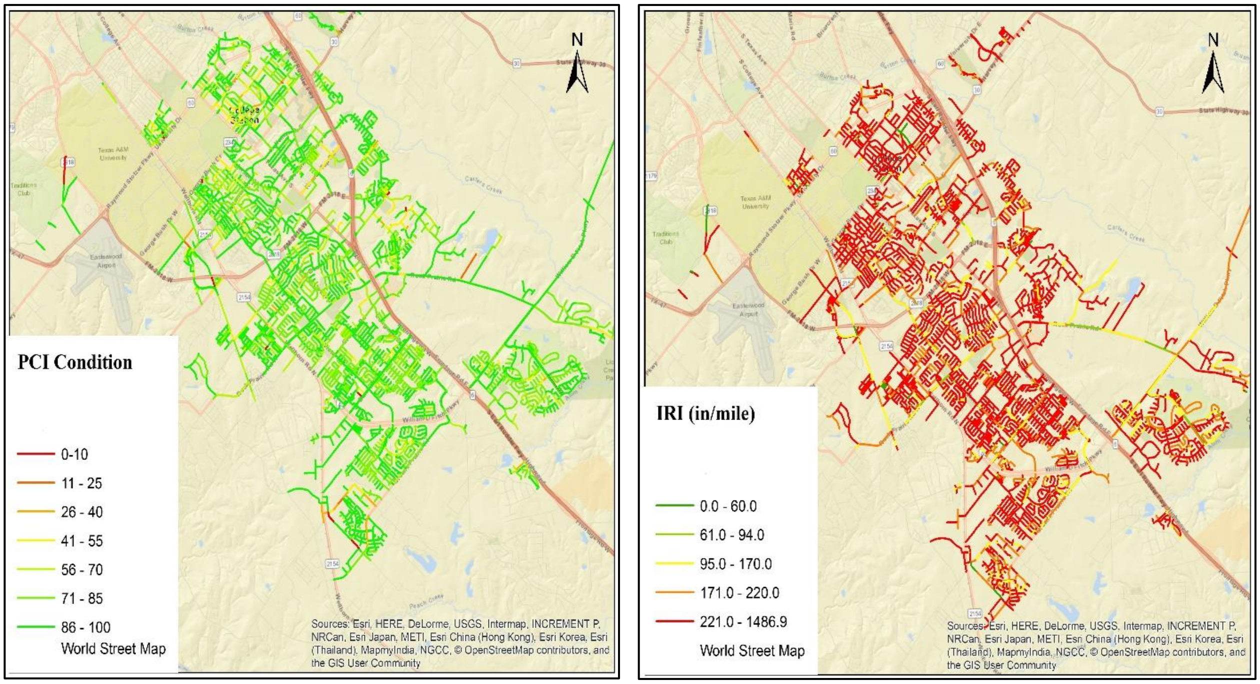
Both PCI and IRI distributions have negative skewness with average PCI of 83 and average IRI of 298 inch/mile. Standard deviation and Coefficient of Variation (CV) were also calculated for both PCI and IRI which can be seen in Table 3-2. CV is a measure of variability of data relative to their average; and since it is dimensionless, it allows to compare two distributions with different units (Lovie et al, 2005). As it can be seen in Table 3-2, CV for IRI is almost 2.5 times greater than PCI which indicates more dispersion in IRI measurements in comparison to PCI.

Table 3-2: Coefficient of variation of PCI and IRI

Condition Index	PCI	IRI (in/mile)
Average	83	298
Standard Deviation	14	127
Coefficient of variation	0.17	0.43

In the next step, network condition was compared based on PCI and IRI. According to ASTM D6433-11 (ASTM 2011), 95% of the road network is in fair or better condition based on PCI, however, only 13% of road network is in fair or better condition based on IRI. A more detailed look at IRI measurements suggests that about 85% of network is in unacceptable condition based on FHWA classification, however, in reality the network is actually in good condition and PCI data is also an evidence for this claim.

In Figure 3-3, the network condition of sections can be found based on IRI and PCI. While the overall condition of network is in good or very good condition, the same network is categorized as failed based on IRI. Therefore, it can be concluded that choosing a pavement condition index to implement Pavement Management Systems (PMSs) should be with careful considerations



N
PCI Condition
0-10
11 - 25
26 - 40
41 - 55
56 - 70
71 - 85
86 - 100
World Street Map
Sources: Esri, HERE, DeLorme, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), MapmyIndia, NGCC, © OpenStreetMap contributors, and the GIS User Community
N
IRI (in/mile)
0.0 - 60.0
61.0 - 94.0
95.0 - 170.0
171.0 - 220.0
221.0 - 1486.9
World Street Map
Sources: Esri, HERE, DeLorme, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), MapmyIndia, NGCC, © OpenStreetMap contributors, and the GIS User Community

Figure 3-3: Comparison of network condition based on PCI and IRI

regarding the functional classification, type, and traffic of the road network. Otherwise, a PMS can result in completely different M&R plans for the same road network.

3.4 Analysis

In this part, first a correlation analysis is conducted to evaluate the relationship between IRI and PCI. Next, a regression model is developed for each functional classification of road network using the College Station dataset. At the end, previously developed regression models are tested to evaluate their accuracy in estimating PCI from IRI or vice versa.

3.4.1 Correlation Analysis

To assess the correlation of two variables, different correlation coefficients can be used which can elucidate the strength and the direction of correlation. The most common correlation coefficient is called Pearson correlation coefficient that can evaluate the strength of a linear relationship between two variables. To use Pearson correlation coefficient, there are some assumptions that need to be satisfied in the first place. These assumptions are as follows (Hauke et al., 2011):

1. Variables need to be normally distributed.

2. Variables are linearly related.

3. Variables need to be measured on interval scales.

In the first step toward evaluating the correlation of PCI and IRI, the dataset was tested to make sure that variables' distributions do not violate any of the Pearson correlation coefficient assumptions.

Kolmogorov-Smirnov and Shapiro-Wilk tests were conducted for testing the normality of the IRI and PCI distributions. The null hypothesis for both tests assumes that their frequency distributions follow normal distributions. In both cases, the null hypothesis was rejected with a confidence interval of 95%. The results of these two tests are presented in Table 3-3.

Table 3-3: The results of normality tests

	Kolmogorov-Smirnov			**Shapiro-Wilk**		
Variable	**Statistic**	**df**	**Sig.**	**Statistic**	**df**	**Sig.**
IRI	0.63	3877	0.000	0.937	3877	0.000
PCI	0.133	3877	0.000	0.867	3877	0.000

Since the data distribution violates one of the main assumptions of Pearson correlation coefficient—which is the normality of variable distribution—this method cannot be used. Another correlation coefficient that is used when Pearson correlation coefficient assumptions are not satisfied is Spearman's correlation coefficient. Unlike Pearson correlation coefficient, Spearman's correlation coefficient neither requires variables to follow any distribution, nor assumes that variables should be linearly related (Hauke et al., 2011). However, the data should hold the following two assumptions in order to get meaningful results from Spearman's correlation coefficient:

1. Variables should be monotonically related.
2. Variables should be either interval or ordinal.

Both these assumptions are satisfied for PCI and IRI in the College Station database. Therefore, Spearman's correlation coefficient can be used to assess the correlation of IRI and PCI. The result of this analysis can be seen in Table 3-4.

Table 3-4: Spearman’s correlation coefficient results

PCI-IRI	
Correction Coefficient	-0.237
Sig. (2-tailed)	0.000
N	3877

As can be seen based on the Spearman’s correlation coefficient, there is a weak, negative monotonic correlation between IRI and PCI ($r_s = -0.237$, n = 3877, $p < 0.0001$). The small p-value suggests that although the correlation between PCI and IRI is week, there is actually a relationship between these two indices. However, it should be noted that the statistical significance is affected by sample size. By increasing the sample size, p-value can be decreased to fall in the rejection zone of the null hypothesis. Therefore, having a very small p-value alone without considering the effect of sample size can be misleading. Spearman’s Correlation Coefficient itself is also a measure of the effect of sample size, therefore it can be argued that the degree of correlation has little practical significance.

3.4.2 Regression Models

In the next step, a regression model for each functional classification of road is developed to estimate PCI based on IRI. As mentioned before, PCI, IRI, and roadway functional classification data for College Station road network were provided by the City of College Station municipality for 2017. Based on this information, the road network was divided into 6 categories of Alley, Local Street, Minor Collector, Major Collector, Minor Arterial, and Major Arterial.

In order to find the relationship between PCI and IRI for College Station road network, statistical regression models were developed using JMP and Microsoft Excel. Ordinary Least Squares (OLS) method was used to develop the regression models and to determine the parameters in regression models. OLS is a common method since it benefits from straightforward calculations and it can compute the minimum sum of squares of the difference between estimates and observations. The statistical significance of each regression model was evaluated using ANOVA (F-test) at 5% level of significance.

After testing different transformations, one regression model was developed for each roadway functional classification using College Station data for 2017. A summary of the models can be seen in Table 3-5.

Table 3-5: Summary of regression models

Functional Classification	Number of Observations	Models	R^2	RMSE	ANOVA	
					F Ratio	P-value
Alley	82	PCI=91.4389-0.0570IRI	0.2417	14.1721	25.4998	0.0001
Local Street	2508	PCI=89.9840-0.0200IRI	0.0341	13.1579	88.3247	0.0001
Minor Collector	531	PCI=90.1055-0.0336IRI	0.0388	16.2792	21.3731	0.0001
Major Collector	464	PCI=95.1153-0.0495IRI	0.0934	15.3658	47.6291	0.0001
Minor Arterial	218	PCI=91.5244-0.0231IRI	0.0199	13.8211	4.3978	0.0371
Major Arterial	46	PCI=98.4578-0.0878IRI	0.0878	10.8103	4.2396	0.0454

As indicated in Table 3-5, the R-squared varies from 0.0199 to 0.2417. In other words, IRI can predict a very low percentage of variation in PCI using linear regression model. RMSE or the square root of the variance of residuals in all roadway functional classifications are large which indicate that the absolute fit of the model is not close to the data and cannot accurately predict the response. The result of F-statistic shows that all P-values are less than 0.05 and therefore, regression coefficients are not equal to zero.

3.4.3 Previously Developed Models

In order to test the accuracy of previously developed models in the literature, these models were used to estimate PCI and IRI using City of College Station database.

Each regression model was used to estimate PCI using IRI and vice versa, then RMSE was calculated for each model to assess the difference between estimated and actual values. In Tables 3-6 and 3-7, the results of RMSE analysis for estimating IRI and PCI are presented. As can be seen, all the models have high RMSE values which indicate that these models cannot be used to estimate PCI using IRI and vice versa.

Table 3-6: Testing previously developed regression models for estimating IRI from PCI

Model	Equation	Number of Observations	Average IRI	IRI Standard Deviation	RMSE
A	IRI = 0.0171(153-PCI)	3877	75.6	15.6	255.6
B	IRI = 18.6-3.41ln(PCI)	3877	229.8	43.9	146.0

Table 3-7: Testing previously developed regression models for estimating PCI from IRI

Model	Equation	Number of Observations	Average PCI	PCI Standard Deviation	RMSE
C	Log(PCI) = 2 - 0.436Log(IRI)	3877	53.9	11.0	33.6
D - Arterials	PCI = -0.206(IRI) + 114.15	264	71.7	16.9	24.98
D - Collectors	PCI = -0.217(IRI) + 115.32	995	60.0	20.9	31.89
D - Locals	PCI = -0.186(IRI) + 110.31	2590	51.2	21.4	39.32

To have a better understanding on the magnitude of RMSE, the average RMSE of PCI and IRI estimation models can be compared with PCI and IRI classification standards. An average RMSE of 32 PCI can result in a shift in the total condition of network from satisfactory to poor. The same results can be also seen for IRI. While the difference between poor and very good classification is only 160 in/mile, having an average RMSE of 220 in/mile means that these models cannot describe the condition of network with enough accuracy; and it can be concluded that these models cannot give any meaningful estimations.

The same results can be seen in Figure 3-4. While in an ideal estimation model, the predicted and actual values should be around the bisector, red line, but this pattern cannot be observed in these models.

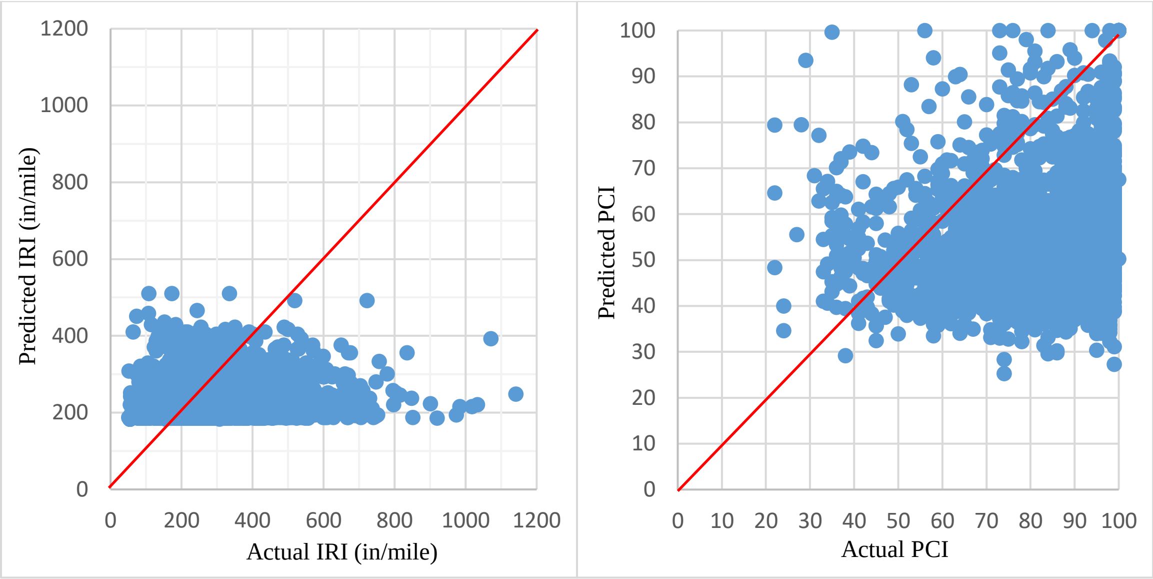


Figure 3-4: Predicted versus actual variables

3.5 Summary

In the recent years, more number of small municipalities have started measuring IRI for their road network without considering the validity of these measurements. The comparison of condition of the road network based on PCI and IRI indicates that although 95% of road network is in fair or better condition based on PCI, 85% of road network is in unacceptable condition based on IRI. This result is a proof that using IRI as a pavement condition indicator in urban areas can lead to misleading information about the condition of the road network.

Additionally, several studies have suggested that there is a meaningful relationship between IRI and PCI. The existence of a meaningful relationship between IRI and PCI in urban road networks can be a strong reason to justify measuring IRI in urban areas. However, in this study, a weak correlation was found between these two variables. The lack of a meaningful relationship between these two pavement performance indicators in urban areas can be explained by distortions of IRI measurements in local road networks. Finally, it can be concluded that IRI may not be a suitable measure of pavement condition for urban streets due to external distortions. If local agencies allocate their resources to improve the methods they use to measure PCI or other condition indices which are originally developed for urban areas, they can get better and more accurate results.

4 DEVELOPMENT OF A METHOD FOR CALIBRATING PAVEMENT PERFORMANCE PREDICTION MODEL FOR AREAS WITHOUT HISTORICAL CONDITION DATA

4.1 Introduction

Pavement performance prediction models are an essential part of any PMS regardless of the modeling method. These models can provide valuable information for different aspects of a pavement management system, such as:

- Forecasting the future condition of the road network;
- Performing life-cycle cost analysis;
- Estimating the approximate time for maintenance and rehabilitation
- Evaluating the benefits of each M&R scenario in a long-term M&R plan.

Prediction models are broken into two main types: probabilistic and deterministic. Probabilistic models such as survival analysis, Markovian processes, Bayesian models, and Monte Carlo simulations are used to assign probability distributions to outcomes and their occurrences. On the other hand, deterministic models, which are divided into mechanistic, empirical, and mechanistic-empirical models, give expected value of the predicted variable. Mechanistic models are based on pavement response parameters, however empirical models are based on extensive historical data and curve fitting, while mechanistic-empirical models are a combination of both.

Over the past four decades, many studies have been conducted to investigate different methods for developing accurate prediction models. Various approaches have been suggested in these studies; however, almost all of the studies share one common requirement in order to develop a performance prediction model: the need to obtain historical condition data of the road network.

In this study, a method for calibration of performance prediction models without historical condition data is suggested.

4.2 Methodology

Changes in the condition of the road from one year to the next year can be expressed as Equation 4-1. In this equation, PCI of each year can be predicted by knowing the PCI of the previous year, and the total change of condition due to deterioration and M&R treatments. PCI_D , or loss of PCI due to deterioration, can be found using deterioration models. In this research, the goal is to develop deterioration models, given PCI_D. Therefore, by changing Equation 4-1 to Equation 4-2, $\overline{PCI}_D$ can be calculated.

$$\overline{PCI}_{x+1} = \overline{PCI}_x + (\overline{PCI}_{MR} - \overline{PCI}_D) \qquad \textit{Equation 4-1}$$

$$\overline{PCI}_D = (\overline{PCI}_x - \overline{PCI}_{x+1}) + \overline{PCI}_{MR} \qquad \textit{Equation 4-2}$$

$$\overline{PCI}_D = \Delta PCI + \overline{PCI}_{MR} \qquad \textit{Equation 4-3}$$

$\overline{PCI}_{x+1}$: Average condition of the road network in year $x+1$

$\overline{PCI}_x$: Average condition of the road network in year x

$\overline{PCI}_{MR}$: Average gain in PCI due to maintenance and rehabilitation

$\overline{PCI}_D$: Average loss of PCI due to deterioration

ΔPCI: Change in PCI for one year

In the following, each component of Equation 4-2 is discussed in more depth.

ΔPCI: As explained above, ΔPCI represents the change in PCI during two subsequent years. The main problem is that many of the municipalities do not have historical condition data to calculate this value. In this study, ΔPCI was set to zero to indicate maintaining network condition at constant level throughout the years.

PCI_{MR}: Each year, municipalities develop an M&R plan based on their annual budget to improve the condition of their road network. The budget is usually not enough to cover the whole expenses associated with M&R needs. Therefore, an optimization model is required to prioritize projects which can make the most improvement in the condition of the road network, at the minimum cost. In this research, the PMS method which was developed by Narciso (2013) is used to calculate the increase in condition of the road network due to M&R treatments. Having a pavement performance prediction model is necessary to reach optimization through calculating the costs and benefits of each M&R treatment.

In this study, the general form of the PCI deterioration model that is used is shown in Equation 4-4. In this equation, PCI_0 is the PCI right after the M&R treatment, Age is the number of years from the last treatment, ρ is the slope factor, and β is the prolonged factor. Figure 4-1 shows an example of using this empirical model with different values of ρ and β for different M&R types. As it can be seen in the figure 4-1, the model has a different type of behavior for different types of rehabilitation. For example, PM deteriorates with a steeper slope in comparison to HR, while LR and MR have behaviors that are between the two extremes.

$$PCI_i = \begin{cases} PCI_0 & when\, Age_i = 0 \\ PCI_0 - PCI_0 e^{-\left(\frac{\rho}{Age_i}\right)^{\beta}} & when\, Age_i > 0 \end{cases} \qquad \textit{Equation 4-4}$$

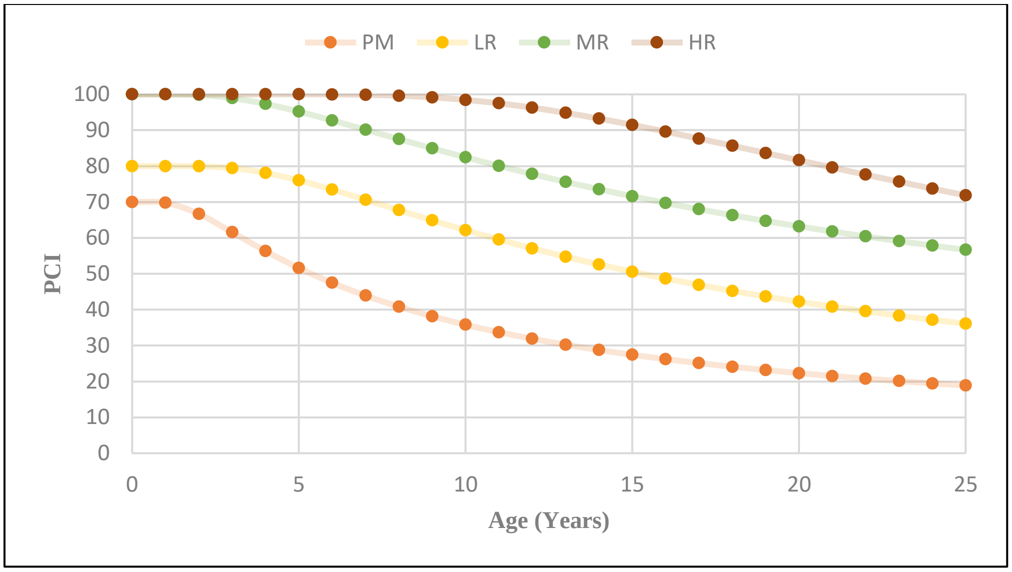


Figure 4-1: An example of PCI deterioration model for PCC pavement

In the next step, an iterative process is followed to estimate the calibration coefficient (ρ and β) that lead to $\Delta PCI = 0$ by repeating this process and using the new set of calibration coefficients, a more accurate performance prediction model is generated. This procedure is illustrated in Figure 4-2.

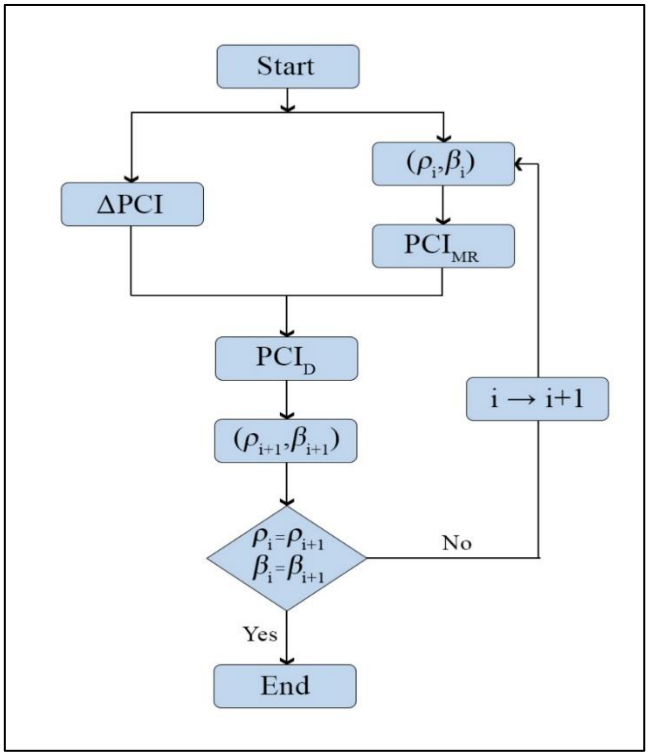


Figure 4-2: Performance prediction estimation methodology

4.3 Project Prioritization Method

The PMS methods used in this study are based on Narciso et al. (2013). In this method, 5 different treatment types are considered. These treatments are presented in Table 4-1.

Table 4-1: Effect of treatments on the pavement

Treatment type	Gain in PCI
Preventive Maintenance (PM)	add 10 PCI
Light Rehabilitation (LR)	add 20 PCI
Medium Rehabilitation (MR)	Reset PCI to 100
Heavy Rehabilitation (HR)	Reset PCI to 100
Do Nothing (DN)	0

For each alternative, the initial costs, the cost of the required treatments over the life-cycle, and the salvage values were also calculated. The costs were calculated in present value using Equation 4-5.

$$PV = Initial\ Expenditure + \sum_{k=1}^{j} Future\ Expenditure \left[\frac{1}{(1+i)^{n_k}}\right] \qquad Equation\ 4\text{-}5$$

Where:

i = Discount Rate

n = Number of years to the year of expenditure

k = Total number of cost items used in the analysis, k goes from 1 to j

For calculating the cost, a period of 30 years was assumed to evaluate the long-term effect of each treatment.

In the next step, the benefit associated with each treatment is calculated. The benefits for each treatment and each section were quantified using the vehicle-miles traveled (VMT) equation which is represented by Equation 4-6.

$$VMT = AUPC \times Length \times AADT \times 365 \qquad \textit{Equation 4-6}$$

Where:

AADT = Annual average daily traffic measured in vehicles/day

Length = Section length measured in miles

AUPC = Area under performance curve measured in years

AUPC could be measured by plotting PCI curves along the considered life-cycle and subtracting the PCI threshold value from the PCI of a given year and summing them over the whole life-cycle. Equation 4-7 shows the formula used to calculate AUPC.

$$AUPC = \int_{i=1}^{n} (PCI_i - PCI_{min}) = \sum_{i=1}^{n} (PCI_i - PCI_{min}) \qquad \textit{Equation 4-7}$$

The AUPC value is affected by the treatment type and lifecycle being considered.

In this study, all viable M&R alternatives are compared to each other and ranked based on their incremental Benefit-Cost (IBC) ratio. The IBC analysis maximizes the total benefit for the whole network at a given budget. This approach identifies the projects that provide the greatest increase in benefit for the least increase in cost. The difference in benefits *(ΔB)* and the difference in cost *(ΔC)* for the consecutive M&R alternatives were computed and the differential BC ratio

($\frac{\Delta B}{\Delta C}$) was calculated. All the negative ($\frac{\Delta B}{\Delta C}$) ratios were removed from the analysis. Finally, the IBC was calculated based on Equation 4-8.

$$IBC_n = \frac{\Delta B_{i-1} + \Delta B_i}{\Delta C_{i-1} + \Delta C_i} \qquad \textit{Equation 4-8}$$

With this approach, the sections with the highest traffic and lowest initial costs are ranked on top of the list since these projects produce the greatest amount of benefit at low costs. On the other hand, the sections that have low daily traffic and require high costs are the last to be executed since they provide the lowest IBC.

4.4 Case Study

The city of College Station is located in Brazos County, Texas. The population of the city is around 110,000 people as of September 2017 (U.S. Census Bureau, 2012). The total length of the road network is 326.46 miles which has been divided into 3877 data collection sections. Each section has a slightly different length with an average length of 0.084 miles. For each section, a variety of information is available such as PCI, IRI, number of lanes, functional classification, AADT, etc. The average PCI for the network without considering the different length of each section is about 83. However, when calculating the weighted average of PCI considering the length of each section, the average PCI for the network is equal to 82; which suggests that the network is in Very Good condition according to the scale provided in ASTM D 6433-11 (ASTM 2011). As can be seen in Figure 4-3, more than 39.4 percent of the sections have a PCI greater than 90 and more than 83 percent of the sections have a PCI greater than 70.

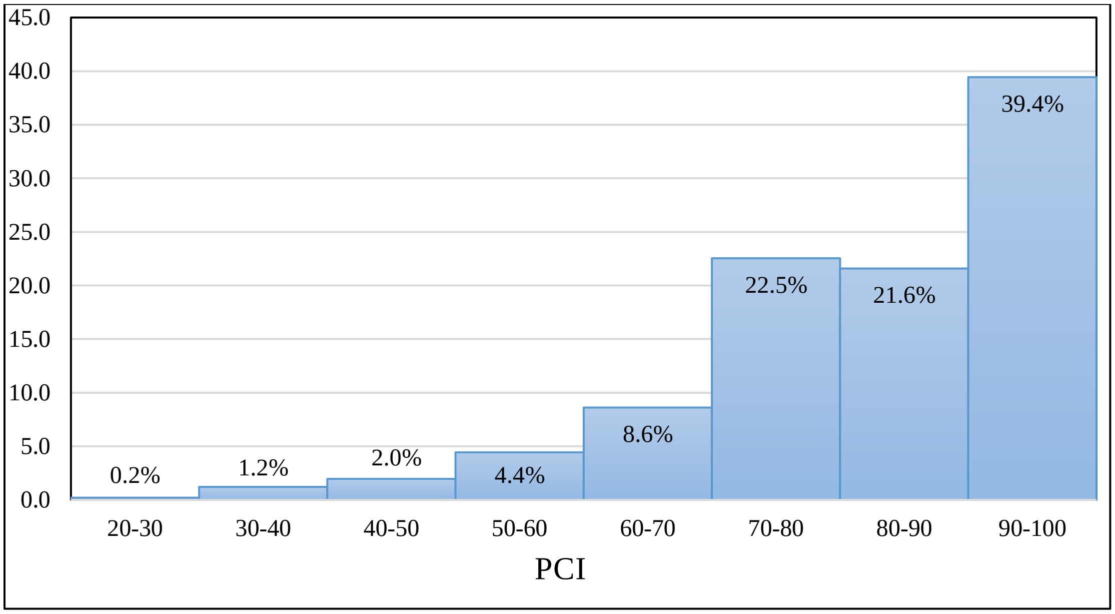


Figure 4-3: Distribution of PCI in the road network

4.5 Model Development

The annual M&R budget for College Station's road network is approximately 5 million dollars per year. To keep the network at its' current average condition, the coefficients of deterioration formula should be calculated in a way that the average loss of PCI due to deterioration is equal to the average gain in PCI caused by the annual M&R. To elaborate, in the first step the increase in PCI for the 5 million dollar budget was calculated. Then, the loss of PCI was calculated by assuming that the change in PCI due to deterioration should be equal to the gain in PCI. In the following, each section has been explained in detail.

The PMS method that was explained before was used to develop a 1-year M&R plan to find out what type of treatment and how much of it is used in one year. The goal of the optimization process is to maximize benefit to cost, or in other words, to provide maximum gain in PCI with minimum cost. The unit cost of each treatment is presented in Table 4-2 which has been derived from the average cost of different types of treatments in Texas (Narciso 2013).

Table 4-2: M&R treatment unit cost

Treatment Type	Unit Cost ($/Lane-Mile)
PM	14,728
LR	76,086
MR	78,429
HR	133,776

The initial parameters which were used in deterioration models were from another study by Narciso et al. (2013). The parameters for different types of pavement family, traffic class, and equivalent single load axle (ESAL) are presented in Table 4-3.

Table 4-3: Initial ρ and β Coefficient for PCI prediction models

			HR		MR		LR		PM	
Pavement Type	**ESAL Class**	**Traffic Speed**	ρ	β	ρ	β	ρ	β	ρ	β
A	High	High	14.8	1.6	10.1	55	8.1	74	6.1	62.7
A	High	Low	14.8	1.6	10	6.8	8.3	8.3	6.5	5.4
A	High	Med	14.8	1.6	10.1	32.7	8.1	32.2	6.1	62.7
A	Low	High	16.2	2.5	12.9	2.4	10.6	2.7	6.9	10.7
A	Low	Low	16.2	2.5	12.9	2.4	10.9	2.4	7.5	4.7
A	Low	Med	16.2	2.5	12.9	2.4	10.8	2.5	7	9.3
A	Med	High	13.2	33.8	11.2	21.4	8.1	85.6	6	62.2
A	Med	Low	13.5	11.1	11.5	3.4	8.4	10	6.1	4.4
A	Med	Med	13.3	28.4	11.3	17.6	8.1	85.6	6	62.2
B	High	High	15.2	2.6	14.4	2.3	14	2.1	7.3	13.5
B	High	Low	15.2	2.6	14.4	2.3	14.2	2.1	8.2	5.3
B	High	Med	15.2	2.6	14.4	2.3	14.1	2.1	7.5	11.9
B	Low	High	13	73.3	9.4	34	7.1	86.6	6.1	73.1
B	Low	Low	13.2	15	9.9	9.7	7.7	7.4	6.6	5.9
B	Low	Med	13	40.6	9.5	28.6	7.3	28.8	6.2	37
B	Med	High	15.2	2.6	14.4	2.3	14	2.1	7.3	13.5
B	Med	Low	15.2	2.6	14.4	2.3	14.2	2.1	8.2	5.3
B	Med	Med	15.2	2.6	14.4	2.3	14.1	2.1	7.5	11.9
C	High	High	19.3	46.4	10.9	26	8.3	16.5	5.5	11.6
C	High	Low	27.2	1.7	11.5	9.4	9	6.8	6.1	3.4
C	High	Med	19.3	38.4	11	19.9	8.5	14.2	5.6	10.2
C	Low	High	12.2	23.7	10	18.3	7.8	34.8	6.3	17.2
C	Low	Low	13.1	7.2	11	6.8	8.4	5.3	7.1	4
C	Low	Med	12.3	20.3	10.1	15	7.7	16.2	6.4	15.2
C	Med	High	19.3	46.4	10.9	26	8.3	16.5	5.5	11.6
C	Med	Low	27.2	1.7	11.5	9.4	9	6.8	6.1	3.4
C	Med	Med	19.3	38.4	11	19.9	8.5	14.2	5.6	10.2

Using Narciso 2013 PMS method, the lane-mile of each treatment that is going to be constructed during one year was computed and is presented in Table 4-4. For calculating the gain in PCI due to the proposed treatments, lane-mile-PCI of each treatment has been calculated for both before and after the treatment. By subtracting the sum of lane-mile-PCI before the treatment from the sum of lane-mile-PCI after the treatment and dividing that number by the sum of lane-mile PCI of the whole network, gain in PCI due to the treatments was calculated as 5.2 (Table 4-5)

Table 4-4: Summary of the optimization

Treatment	Lane-Mile	Average PCI before the treatment	Average PCI after the treatment
PM	150.6	76.4	86.4
LR	33.8	72.4	92.4
MR	44.0	56.9	100.0
HR	0.0	-	-

Table 4-5: Summary of PCI gain calculations

Treatment	Lane-Mile-PCI before treatment (A)	Lane-Mile-PCI after treatment (B)	Total Lane-Mile of the network (D)	B-A	Gain in PCI ((B-A)/D)
PM	11508.1	13014.2	788.5	1506.1	1.9
LR	2446.4	3122.4	788.5	676.0	0.9
MR	2503.9	4402.0	788.5	1898.1	2.4
HR	0.0	0.0	788.5	0.0	0
				Total	5.2

As mentioned before, to find out the parameters of Equation 4-2, the loss of PCI due to this equation should be equal to the gain of PCI that has been calculated in the previous part, so the average PCI of the whole network would remain constant. Equation 4-4 needs to be solved to find ρ and β. In addition, the available PCI values are for 2017, however the age of each section from the time of last major treatment should be calculated first. An Excel sheet was developed to solve

Equation 4-4 for ρ and β. Since there are two variables and only one equation, there are more than one set of answers for the equation. However, by setting boundaries based on the maximum and minimum possible values of ρ and β, the viable answers would be reduced to less than 10 sets of solutions. To solve this equation, the "Evolutionary" method in Solver is used since it is the best method in Excel to solve non-linear problems despite the prolonged calculation time in comparison to other available methods. The viable range for the slope factor usually changes between 6 to 16. By fixing ρ for every integer in this range, β is found for each ρ. The results are shown in Table 4-6.

Table 4-6: The pair of best fitted answers for each ρ

ρ	β	Network Average PCI	Standard Deviation
6.0	1.6	73.0	14.1
7.0	1.7	75.2	14.9
8.0	1.6	76.4	15.2
9.0	1.6	76.9	14.8
10.0	1.8	77.1	14.8
11.0	2.2	77.0	15.0
12.0	2.4	77.0	15.0
13.0	2.7	77.0	15.4
14.0	2.8	76.9	15.3
15.0	3.1	76.9	15.3
16.0	3.3	76.9	15.4

As mentioned before, the goal of-these calculations is to find a pair of ρ and β which would result in average PCI of 77.07 after one year. In addition, to have a similar distribution, it is of great importance that the PCI distribution has a similar standard deviation as the previous year. The standard deviation of data from 2017 is equal to 14.4351. Among all pairs of ρ and β, only two of them give close enough answers: ρ = 9, and ρ = 10. The performance prediction models with ρ equal to 9 and 10 can be seen in Table 4-7 and Figure 4-4.

Table 4-7: The pair of solution for the first iteration

ρ	β	PCI	Standard Deviation
9.0	1.6	76.9	14.8
10.0	1.8	77.1	14.8

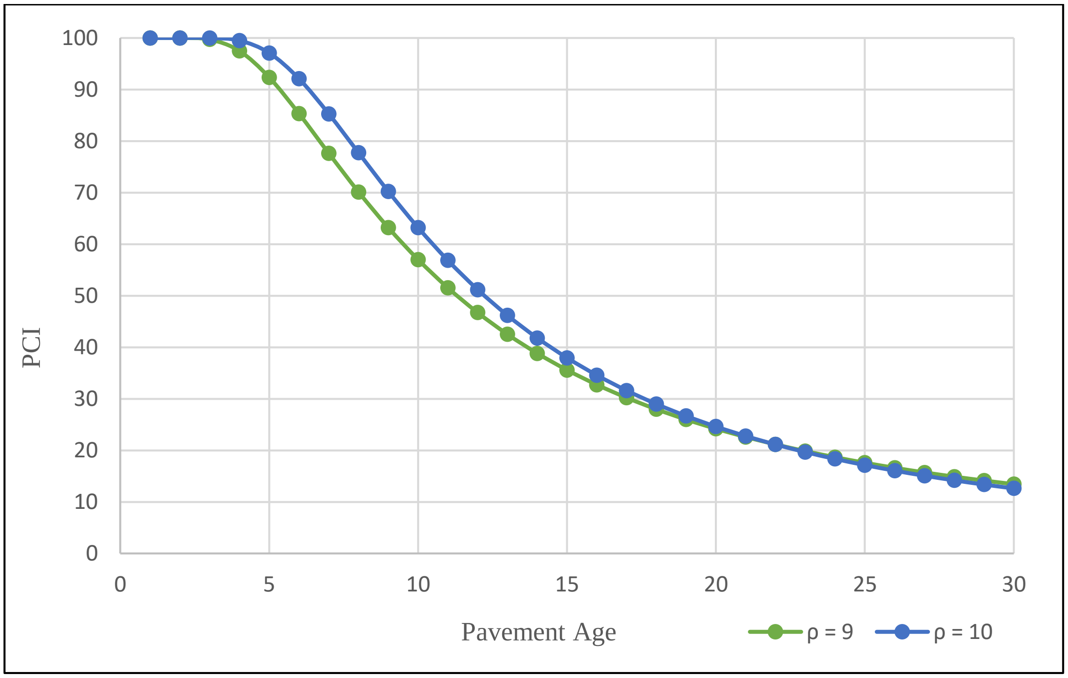


Figure 4-4: Prediction curved for $\rho = 9$ and $\rho = 10$

In order to develop a PMS, a pavement prediction model should consider different M&R types, so that the effect of spending more money on comprehensive treatments can be seen in calculation of cost and benefit. Otherwise, the cheapest treatment will always be selected due to lower cost and equal benefit to other treatments. The upper and lower limits of ρ and β for each treatment from City of Bryan PMS which is shown in Table 4-8 is used as boundaries for finding a pavement prediction model for each treatment and the results are shown in Table 4-9 and Figure 4-5.

Table 4-8: The upper and lower limits of ρ and β for each treatment from Bryan PMS

		Minimum	Maximum	Average
HR	ρ	12.2	27.2	16.0
	β	1.6	73.3	16.8
MR	ρ	9.4	14.4	11.7
	β	2.3	55.0	14.1
LR	ρ	7.1	14.2	9.8
	β	2.1	86.6	21.1
PM	ρ	5.5	8.2	6.6
	β	3.4	73.1	20.3

Table 4-9: ρ and β for different treatments

	ρ	β
PM	7.0	1.7
LR	9.0	1.6
MR	11.0	2.2
HR	14.0	2.8

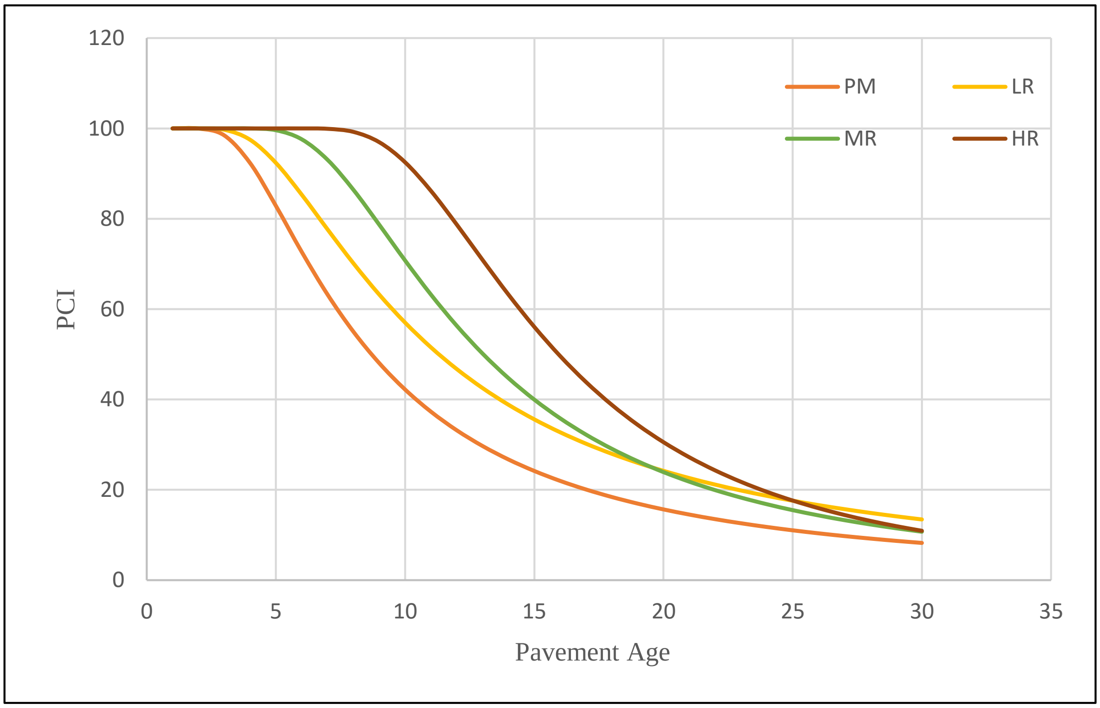


Figure 4-5: PCI deterioration models for different M&R treatment types

In the next step, by using calculated values of ρ and β, the same procedure is repeated and a new set of ρ and β is calculated. A new M&R treatment plan is developed using the new set of coefficients, and PCI gain due to the new M&R plan is calculated. Then, by repeating the same procedure a new ρ and β is calculated. The results can be seen in Table 4-10, 4-11, and 4-12.

Table 4-10: The summary of the optimization in second iteration

Treatment	Lane-Mile	Average PCI before the treatment	Average PCI after the treatment
PM	104.0	76.4	86.4
LR	33.8	72.4	92.4
MR	44.0	56.9	100.0
HR	0.0	-	-

Table 4-11: The summary of PCI gain calculations in second iteration

Treatment	Lane-Mile-PCI before treatment (A)	Lane-Mile-PCI after treatment (B)	sum of the Lane-Mile of the network (D)	B-A	Gain in PCI ((B-A)/D)
PM	7946.6	8986.6	788.5	1040.0	1.3
LR	3857.8	4923.8	788.5	1066.0	1.3
MR	4049.9	7120.0	788.5	3070.1	3.9
HR	0	0	788.5	0	0
				Total	6.6

Table 4-12: ρ and β for different treatments in second iteration

	ρ	β
PM	7.0	1.7
LR	9.0	2.1
MR	11.0	2.6
HR	14.0	3.4

This procedure should be repeated until the coefficients converge. In this example, only two iterations is needed so that coefficients converge. The same performance prediction models are resulted by using the coefficients in second iteration. Therefore, the procedure is stopped at the end of second iteration.

4.6 Sensitivity Analysis

The cost of each treatment per lane-mile is presented in Table 4-4 based on average data found in the literature. However, since the cost of treatment has direct effects on the results of the model, a sensitivity analysis is conducted to compare the effect of changing unit cost on the values of ρ and β. To do so, two scenarios with $\pm 20\%$ of the current unit cost are considered. Unit cost of each treatment for all three scenarios can be seen in Table 4-13.

Table 4-13: Unit cost of each scenario ($/Lane-Mile)

Treatment	Original	+20%	-20%
PM	14,728	17,674	11,782
LR	76,086	91,303	60,869
MR	78,429	94,115	62,743
HR	133,776	160,531	107,021

For each scenario, optimization was performed to obtain a 1-year M&R plan and the results are shown in Table 4-14. As can be seen, as the unit cost increases, the total lane-mile decreases considering a fixed maximum annual budget of $5 million. A smaller M&R plan causes a smaller increase in PCI that can affect the values of ρ and β.

Table 4-14: M&R plan for each scenario

Treatment	Original	+20%	-20%
PM (Lane-Mile)	104	82.5	110.5
LR (Lane-Mile)	53.3	56.2	60.7
MR (Lane-Mile)	71.2	54	84.7
HR (Lane-Mile)	0	0	0
Total (Lane-Mile)	228.5	192.7	255.9
Increase in PCI	6.6	5.4	7.6

ρ and β are calculated for each scenario which can be seen in Table 4-15, Figure 4-6, and Figure 4-7. As it is shown, unit cost can cause a noticeable change in the values of ρ and β. ρ and β are constant regardless of unit cost for PM treatment, however, for other treatments, unit cost can have a notable effect on values of ρ and β.

Table 4-15: Values of ρ and β for different Scenarios

Treatment	**Original unit cost**		**+20% unit cost**		**-20% unit cost**	
	$\boldsymbol{\rho}$	$\boldsymbol{\beta}$	$\boldsymbol{\rho}$	$\boldsymbol{\beta}$	$\boldsymbol{\rho}$	$\boldsymbol{\beta}$
PM	7.0	1.7	7.0	1.7	7.00	1.7
LR	9.0	2.1	9.0	1.6	8.00	2.1
MR	11.0	2.6	11.0	2.1	12.00	3.3
HR	14.0	3.4	16.0	3.3	14.00	4.0

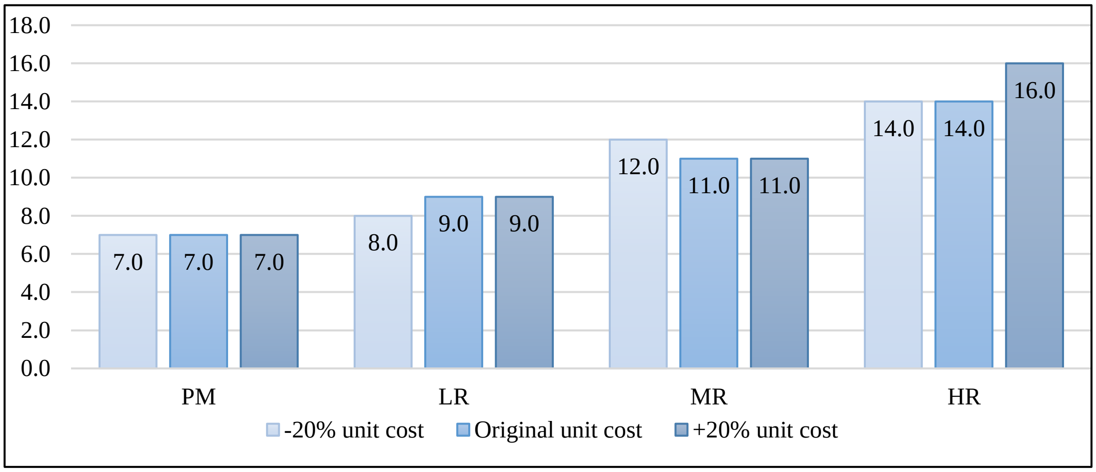


Figure 4-6: Values of ρ for different scenarios

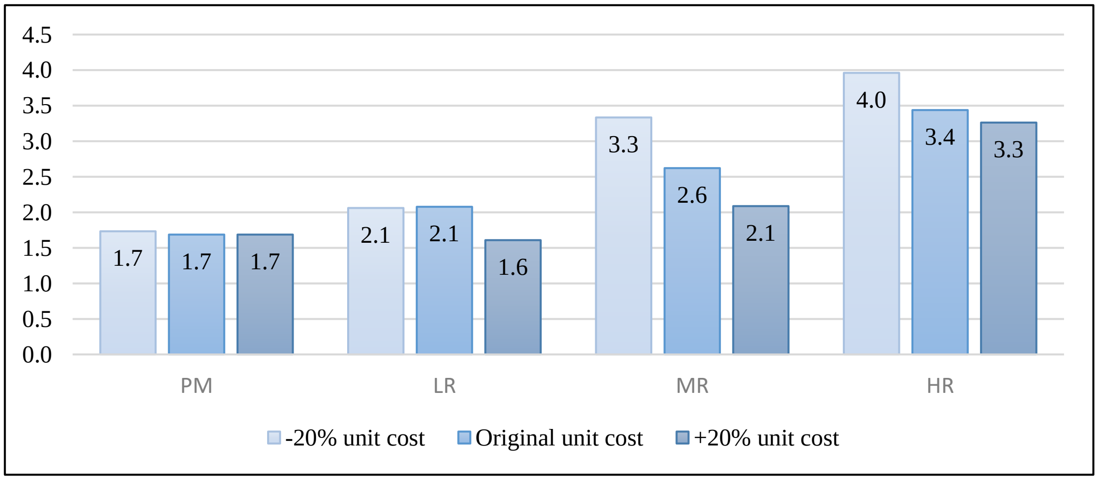


Figure 4-7: Values of β for different scenarios

5 DEVELOPMENT OF A ROADWAY NETWORK SEGMENTATION METHOD IN ARCGIS

5.1 Introduction

In order to prioritize M&R, an inventory of road characteristics should be developed including different types of information such as pavement condition, roughness, annual average daily traffic (AADT), width and length of each section, functional classification, previous M&R, etc. This information is then used to perform network segmentation into management sections, and calculate the costs and benefits of M&R projects (City of Nashville, 2018).

In this research, a segmentation method suitable for urban areas has been developed based on the Proximity to Deficient Areas Approach (PDA). The segmentation method is designed in a Geographic Information System (GIS) platform to enable agencies to visualize and analyze their road networks. In the following, different segmentation methods have been reviewed first, and then, the new method is discussed in details.

5.2 Segmentation Methods

One of the main steps of every PMS is to identify homogenous sections of a road network in order to form segments with similar characteristics. There are three main approaches to perform segmentation of the roads: Fixed Length, Dynamic Segmentation, and Static Segmentation.

In Fixed Length method, the length of sections is fixed and are usually determined by fixed road features such as mileposts and city blocks. Since this approach completely disregards the pavement attributes such as pavement condition and traffic, it cannot create segments with homogenous characteristics (Bennett, 2004).

In Dynamic Segmentation, the segmentation is done every year and it is independent of the previous year's segmentation result. This method conduct segmentation based on the new condition of segments after performing M&R treatments. Therefore, each year new deficient sections are recognized and put together as new segments. The disadvantage of this method is that the sections cannot be tracked over years to evaluate the effects of performed treatments (Narciso et al, 2013).

In Static Segmentation, the segmentation is done in the first year and the segments are constant until the end of the analysis period. This method allows the user to keep a record of each segment for several years. However, it may be unable to define realistic projects based on the current condition of pavement in comparison to Dynamic Segmentation method (Narciso et al, 2013). To solve this problem, segmentation can be done every 3-5 years so that in addition to having a historical database for each segment, realistic projects can be defined as well (Bennett, 2004).

5.3 Cumulative Difference Approach

Cumulative Difference Approach (CDA) was first developed by AASHTO in 1993. This method can provide a simple but efficient algorithm to group homogenous sections into segments along the project length. Although this method is relatively easy to comprehend, manual implementation of this method for a large database can be very complicated and time-consuming (AASHTO, 1993). One of the most important advantages of this method is that this method can use different types of attributes such as Pavement Condition Index (PCI), International Roughness Index (IRI), skid resistance, and generally all types of indices which can quantify the condition of the road to do segmentation.

Since it is much easier to explain CDA graphically, Figure 5-1 is used as an example of CDA with the assumption of having a continuous and constant response value with three intervals along the project length. In Figure 5-1-a, three separate sections with three different response values can be seen. Cumulative area of these sections in a random point like X can be calculated using Equation 5-1 (AASHTO, 1993).

$$A = \int_0^{x_1} r_1 dx + \int_{x_1}^{x} r_2 dx \qquad \textit{Equation 5-1}$$

By using the above equation, the solid line in Figure 5-1-b can be calculated. In order to figure the dash line, it is enough to calculate the overall average value of the project response using Equation 5-2 and then Z_x is simply the difference between the actual cumulative area and the average lines (AASHTO, 1993).

$$\bar{r} = \frac{\int_0^{x_1} r_1 dx + \int_{x_1}^{x_2} r_2 dx + \int_{x_2}^{x_3} r_3 dx}{L_P} = \frac{A_T}{L_P} \qquad \textit{Equation 5-2}$$

In the next step, Z_x is plotted versus distance as can be seen in Figure 5-1-c. Based on Figure 5-1-c, whenever the slope of Z_x changes algebraic signs, the position of the change-point (unit boundary) changes as well.

CDA is still one of the most common methods being used in many projects. However, this method has some flaws as different studies have pointed out. In the following some of the main drawbacks of this method have been discussed in more details.

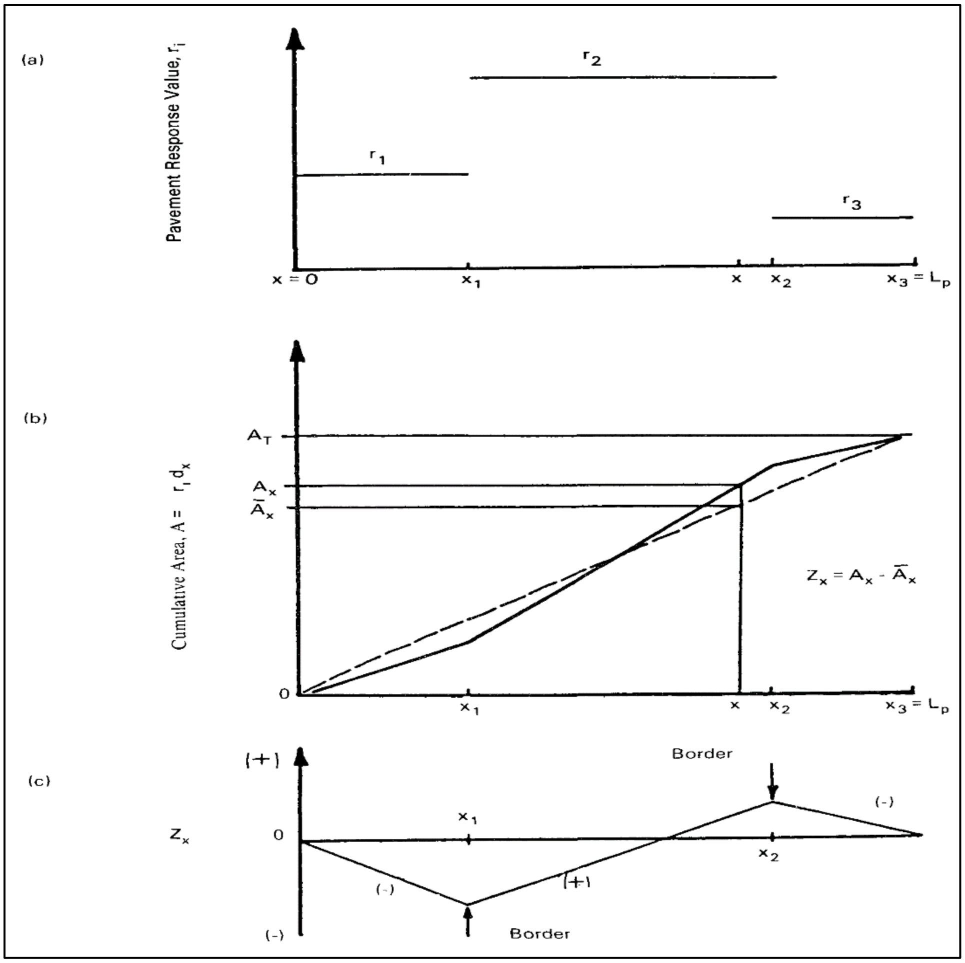


Figure 5-1: Cumulative Difference Algorithm (CDA) (Reprinted from AASHTO 1993)

5.4 CDA Shortcomings

One of the main problems with CDA is that using this method can generate different results based on the starting point of the analysis. In other words, this method is sensitive to the starting and ending points of the analysis and even a slight change in these two points can alter the result of segmentation tremendously (Thomas, 2005).

As explained, the method that is being used to flag segments' borders is contingent on the overall average of response. Meaning that whenever the response curve passes the overall average of response, that point is marked as a segment border. One of the main problems this method can cause is that two consecutive sections with different average responses would be counted as one section as long as both their averages are above or below the overall average of response (El Gendy et al, 2008).

The other problem with CDA is that the user does not have any control on the number of homogenous sections, and minimum or maximum length of each section. Therefore, this method may suggest very short or very long homogenous sections which are not realistic in construction projects (Misra et al, 2003). Additionally, the user cannot specify a threshold so that all the sections below that threshold would be defined as segments in need of M&R treatments.

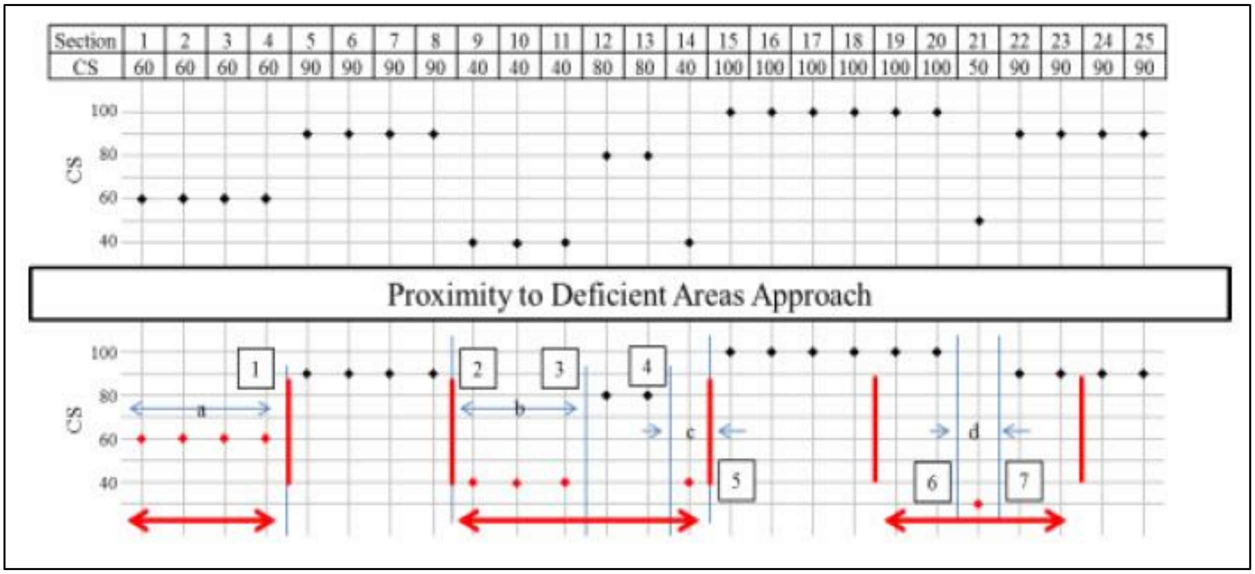
There have been multiple studies to make this method more robust toward road networks with large databases and to automate the segmentation procedure using CDA, in addition to fixing some of the shortcomings of this method. In the following, some of these studies have been discussed in detail.

In 2000, James Kennedy et al. proposed a method for segmentation which joins contiguous sections of the road with similar characteristics. In this method, CDA is used to identify homogenous sections. A script was written using Visual Basic to implement Dynamic Segmentation. By using this code, the user can change the segmentation criteria and also repeat the segmentation to further decrease the number of segments. Additionally, a code was written to implement segmentation for all the sections in a GIS platform. ArcGIS can provide a platform for visualization of data, and also deliver tools to help with decision-making. Trans CAD GIS software was used to connect the segmented data to the respective highway map (Kennedy, 2000).

Cuhadar et al. (2002) developed a new algorithm based on the wavelet transform for segmentation of the pavement network. In the first step, a noise reduction technique was used to remove small ripples. In order to do so, first, the wavelet coefficients of data were computed. Next, those coefficients which were smaller than a threshold were shrunk; and finally, the signal was again created using the inverse wavelet transform. CDA was used as a basis for performing segmentation. Since CDA-based methods are not usually accurate while working with data with noise-like ripples, the noise deduction technique becomes more crucial for a precise segmentation. In the next step, an analysis on the singularities of the condition data was performed to identify segment borders from isolated singularities to create segments. This method can also be combined with GIS to provide a user-friendly visual platform for performing segmentation (Cuhadar, 2002).

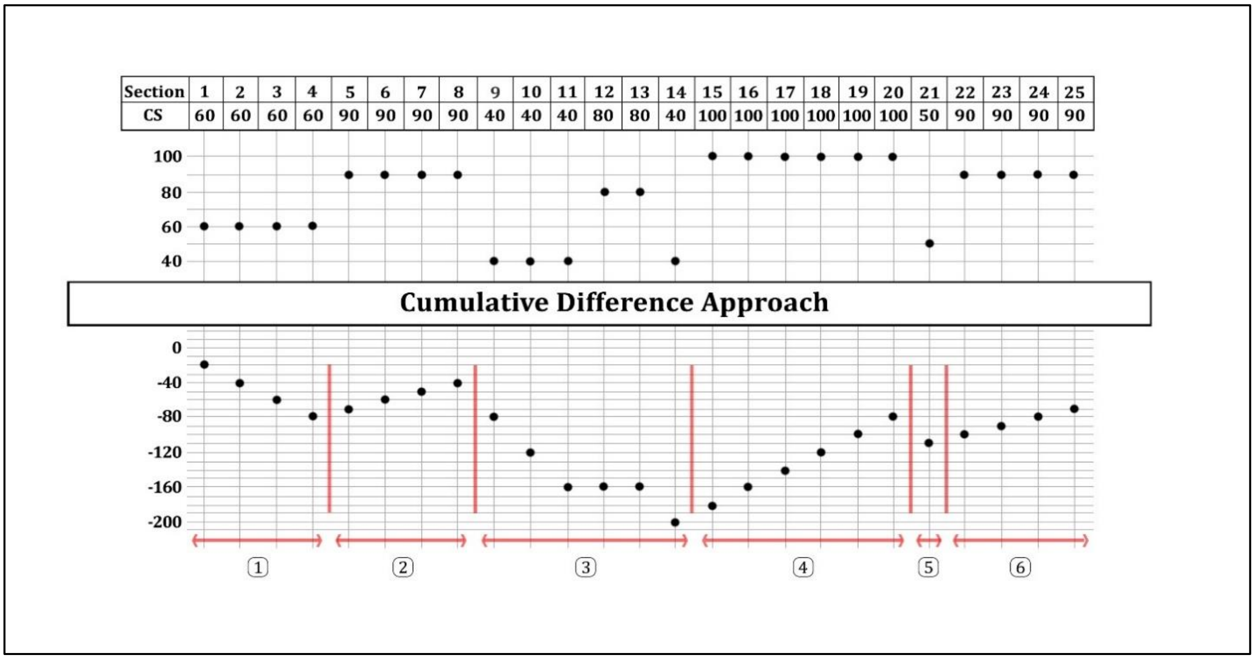
5.5 Proximity to Deficient Areas Approach (PDA)

PDA was first presented by Narciso et al. (2013). In this method, an M&R trigger criterion is used to determine the deficient localized areas and new segments are then formed around the sections with poor condition. To form realistic M&R projects, PDA joins the problematic sections with other problematic sections which are closer than a certain value. Moreover, in this method, the user can specify the minimum and maximum length of each segment to prevent the formation of very short or very long segments. An example of doing segmentation using PDA is shown in Figure 5-2. In this example, M&R trigger value is set to a Condition Score (CS) of 80. Sections with CS less than 80 are selected and the method joins the sections which are less than 2 miles away. The result is shown in Figure 5-2 (Narciso et al., 2013). This method is usually performed annually and it only groups up sections which are deficient. Therefore, in contrary to CDA, the whole network is not divided into segments. Another advantage of this method is its practicality due to considering different restrictions for the length of each segment. To compare the result of segmentation for CDA and PDA, CDA was also used to perform segmentation on the same example and the result is shown in Figure 5-3. As it can be seen, the number of segments in CDA is twice the number of segments in PDA and the reason is that CDA divides the whole network into segments while PDA only forms segments which need M&R treatment. Additionally, in CDA, due to lack of criteria for the minimum length of each segment, very short segments can be found which in the real world are neither practical nor realistic.



Section	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CS	60	60	60	60	90	90	90	90	40	40	40	80	80	40	100	100	100	100	100	100	50	90	90	90	90

Figure 5-2: Proximity to Deficient Areas Approach (Reprinted from Narciso et al., 2013)



Section	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CS	60	60	60	60	90	90	90	90	40	40	40	80	80	40	100	100	100	100	100	100	50	90	90	90	90

Figure 5-3: Cumulative Difference Approach

5.6 Geographic Information System (GIS)

GIS is a computer system capable of connecting geographically referenced information to attribute data. It provides a spectrum of data handling tools that enable the users to store, analyze, display, and manage spatial data (Elhadi, 2009).

A PMS needs a platform to store road network data, retrieve them, and at the end assist decision-makers to evaluate different scenarios. In this way, GIS could be a powerful tool for implementing pavement management systems. In the following, a short review of Dynamic Segmentation tool in GIS, and some of the studies about utilizing this tool for performing segmentation is provided.

Dynamic Segmentation is the procedure of making features with the capability of being shown and analyzed on a map out of linearly referenced data which are usually stored in tables. Dynamic Segmentation has been used extensively to store, manage, and analyze linear features such as rivers, roads, and pipelines. Moreover, Dynamic Segmentation is also a powerful tool to record different attributes along a linear feature. In summary, Dynamic Segmentation can assist with two main issues regarding analyzing data which have been listed below (Dueker et al., 1993):

a) **Linkage** which relates attribute tables to spatial features.
b) **Segmentation** which creates new boundary points based on the defined criteria for a specific application.

In the following, some of the prominent studies which focus on using Dynamic Segmentation as a tool to perform segmentation of roads and highways are presented.

In a paper published by Timothy Nyerges in 1990, different methods of locational referencing and highway segmentation using GIS were reviewed. Locational data determines the geometry of highway and can affect the results of segmentation if the data lacks required accuracy and consistency. Therefore, different transportation organizations have developed several referencing strategies. One of these strategies which is commonly used to assist with pavement segmentation is Control Section Locational Referencing. In this method, a length of highway with the same characteristic as the control section is considered to be homogenous. Control sections' boundaries are from one milepoint to another so that the length of a control section can vary from one-tenth of a mile—which is the shortest distance between two milepoints—to a couple of miles. Additionally, two segmentation methods which are commonly used in GIS are Fixed Length Segmentation that is based on homogenous characteristics of control sections; and Variable-length Segmentation which is based on the continuous observation of a specific characteristic along the highway such as pavement condition until the attribute no longer exists (Nyerges. 1990).

Niraki et al. (2009) assessed the application of Dynamic Segmentation in pavement management. In his research, he proposed a road maintenance data model using Dynamic Segmentation which was implemented on the Iranian Road Network. It was found that using Dynamic Segmentation can improve handling the data in many ways as briefly explained below (Niraki et al. 2009):

1- Because of the logic used to store the attribute tables in Dynamic Segmentation method, the analysis can be performed faster than Arc-node structure.
2- By using Dynamic Segmentation, multiple numbers of attribute tables can be assigned to the linearly referenced attributes.
3- Dynamic Segmentation provides the ability to segment an arc without actually breaking the line.
4- By utilizing Dynamic Segmentation, data can be altered at any time without making any changes in the road network configuration.

5.7 Interurban and Urban Road Network

While segmentation has been done on interurban road networks for years, performing segmentation on urban road networks is a relatively new topic. Interurban road networks mostly consist of linear long roads which connect two different destinations. Moreover, due to transportation restrictions to classify a road as a highway, they have been designed in a way that no or a minimum number of intersections with other roads would be seen which makes the segmentation process relatively straightforward.

On the other hand, automated segmentation of urban road networks can be very complicated due to the existence of very short sections, loops, forks, multiple intersections, and arbitrary connections. To be able to program a script that performs segmentation automatically, a solution should be suggested for each of these situations. Figure 5-4 is an example of how different the urban and interurban road networks could be. As it can be seen on the left, the urban road network consists of multiple short sections, intersections, and loops. However, the interurban road network consists of long roads. In the following, some of the challenges faced while implementing segmentation for the urban road network are listed and discussed in detail.

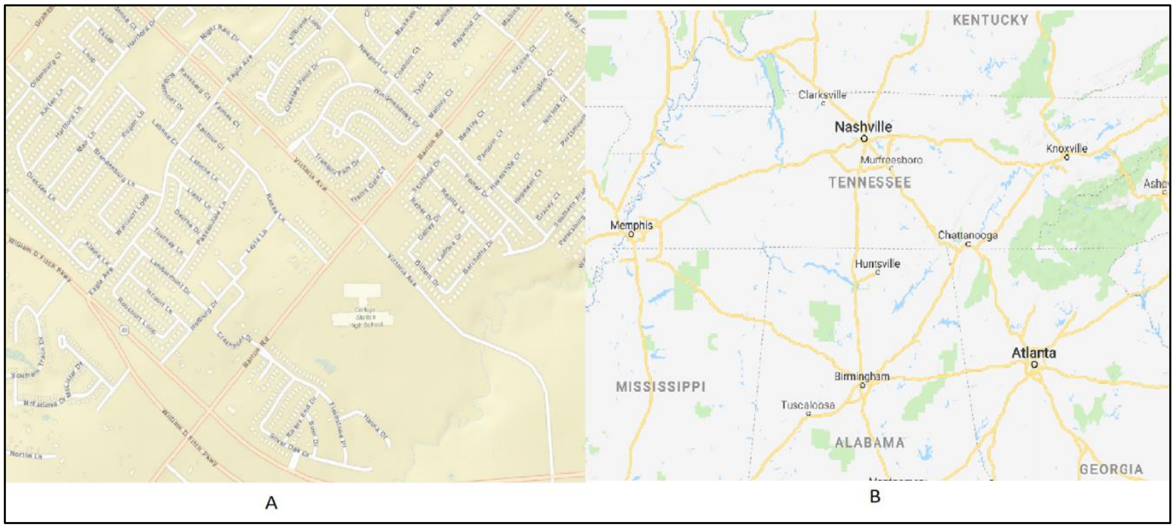
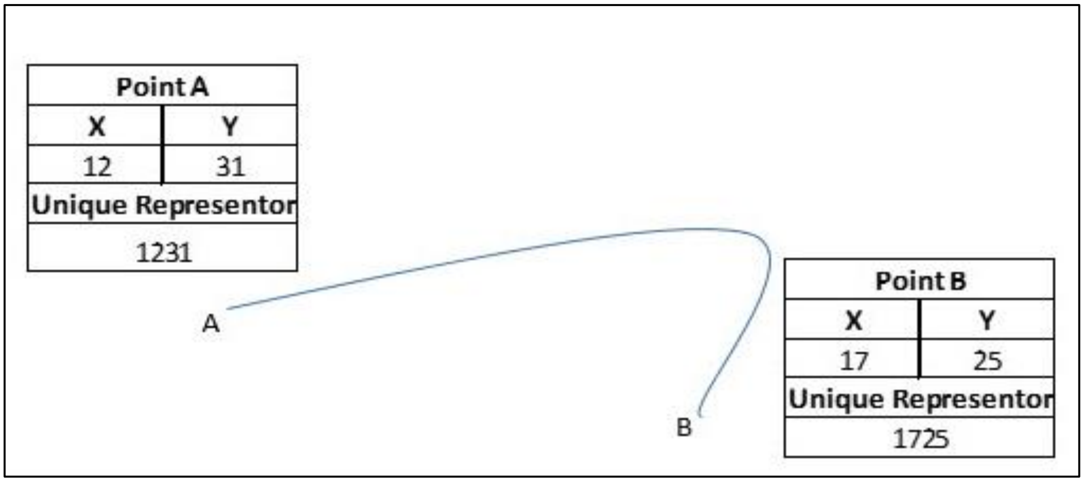


Figure 5-4: Comparison of the urban road network (A) and the interurban road network (B)

In some road network databases, there is no information about the starting and ending point of each section. To obtain this information, function “Add Geometry” in ArcGIS was used to acquire coordinates of starting and ending points of each section. Since the code has been designed in a way that only one number can represent the beginning or the end of a section; X-coordinate and Y-coordinate for each point were combined to reach one unique number. An example of this procedure can be seen in Figure 5-5.

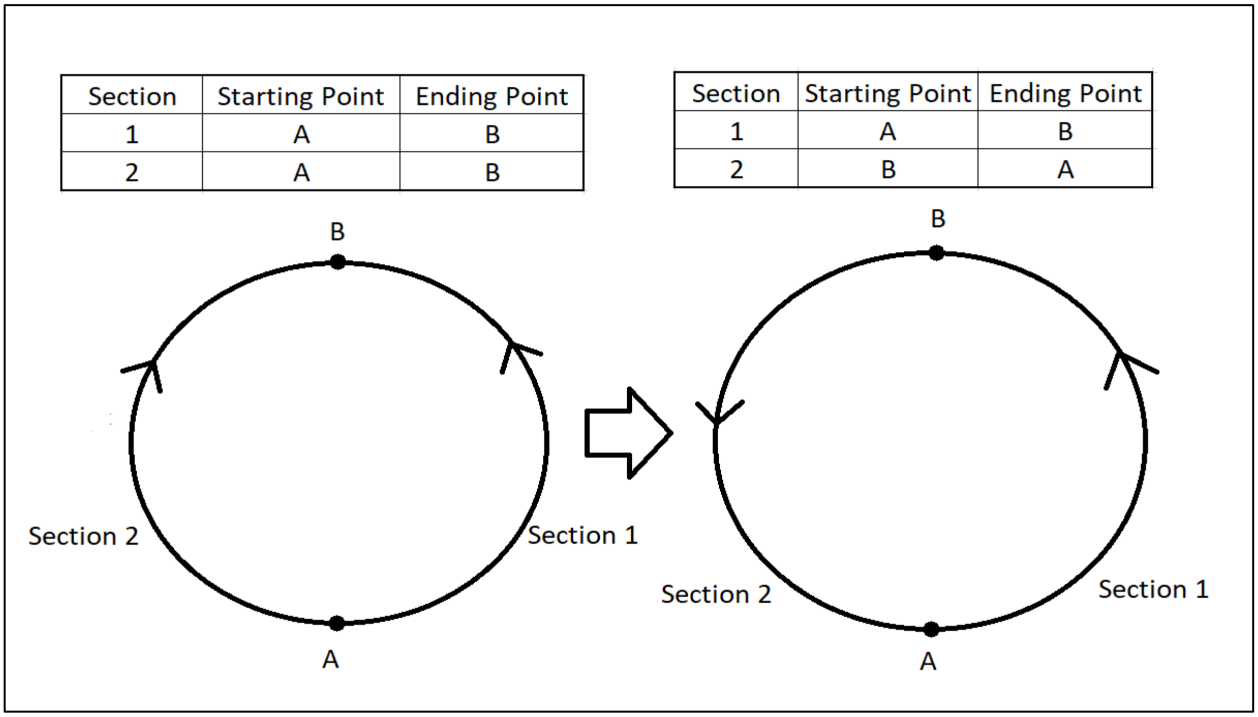


Point A	
X	Y
12	31
Unique Representor	
1231	

Point B	
X	Y
17	25
Unique Representor	
1725	

Figure 5-5: Defining unique representor for each point

The method that is being used to specify each section of the road is performed by using the starting and ending point of each section. Therefore, each section is identified by a unique set of two numbers. A problem which was faced for the first time while performing segmentation for an urban area was that some of the sections had similar starting and ending points and therefore, the code was unable to run properly. This is mainly due to the fact that in some parts of the road network, there are loops which were made of two sections with the same start and end. To solve this problem, a direction is defined for each loop so that the ending of one section would be the starting point of another. In this way, two sections with two different combinations of starting and ending points would be created in the end. Figure 5-6 shows an example of the mentioned problem.



Section	Starting Point	Ending Point
1	A	B
2	A	B

Section	Starting Point	Ending Point
1	A	B
2	B	A

Figure 5-6: An example of defining direction to solve the problem with loop

Another problem occurred while facing fork-shaped streets in an urban road network. The segmentation method has been designed in a way to put continuous sections with the similar street name in one group. Continuity was defined based on whether the ending point of a section is equal to the starting point of the next section. This definition of continuity would work without any problem for simple linear roads. However, in an urban road network, this might not work as desired. An example is when a fork-shaped street appears which has two sections that share similar ending or starting points rather than sharing a point which represents the starting and ending of the other section. An example would help to understand the problem better. In Figure 5-7, All three sections have similar street names so ideally all of them should be in one group. However, only section 1 and section 2 are considered continuous under the definition of continuity and section 3 is neglected since it has a different starting point than point B. To solve this problem, the definition of continuity was changed so that all the sections which share one common point are considered to be continuous regardless of their starting or ending point.

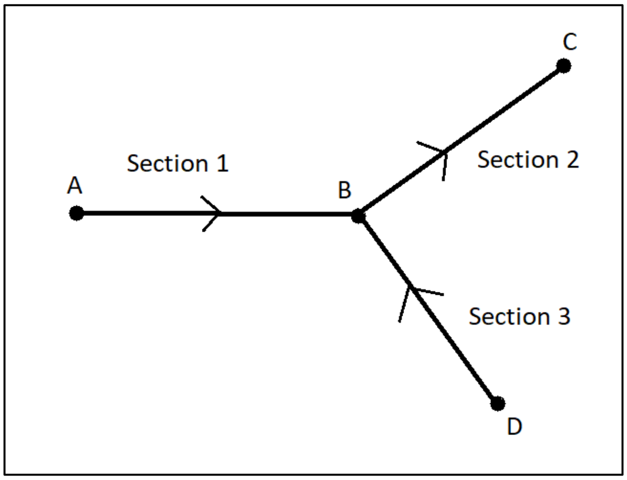


Figure 5-7: An example of problem with fork shape street

5.8 Methodology

In order to develop an automated segmentation method in ArcGIS, first, a database should be prepared with essential information of the road. Three types of data are needed to perform segmentation which are listed below:

a) Pavement Identification: This type of data is needed to specify each section and to identify adjacent sections in order to form continuous segments. The minimum information needed to do so is two numbers to represent the starting and ending points of each section.

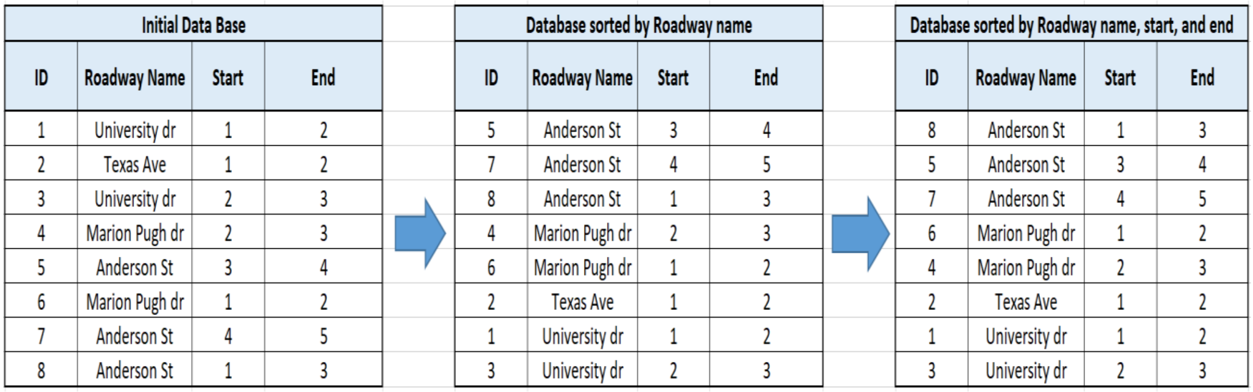
b) Pavement Characteristic: Pavement characteristic data is used in this method to identify homogenous sections, group sections with similar characteristics, and make M&R projects. The minimum data needed for this phase of the project are: Pavement Condition Index (PCI), length of each section, surface type, and roadway name.

c) General Information: This type of information is used to give perspective about each section and they are not necessary for performing segmentation. Information such as speed limit, functional classification, previous treatment, Annual Average Daily Traffic (AADT), number of lanes, IRI or any other type of condition index can be put in this category.

After developing a database with required information to do segmentation, homogenous sections should be identified and grouped together. The process of finding adjacent sections of the road network is called “grouping” which is explained step by step in the following:

1. The database should be sorted in an ascending order based on roadway name. Then, the sorted database should be sorted again ascendingly based on the starting and ending points of sections. This process is illustrated in Figure 5-8.
2. A script is written in Python to go over the sorted database from beginning to the end and identify all the sections that have the same roadway name and surface type and share a mutual starting or ending point. The Python code is shown in Appendix A. An example of this process is shown in Table 5-1.

By implementing these two steps, the road network is going to be divided into groups of adjacent sections and from there, segments in need of a treatment can be identified.



Initial Data Base			
ID	Roadway Name	Start	End
1	University dr	1	2
2	Texas Ave	1	2
3	University dr	2	3
4	Marion Pugh dr	2	3
5	Anderson St	3	4
6	Marion Pugh dr	1	2
7	Anderson St	4	5
8	Anderson St	1	3

Database sorted by Roadway name			
ID	Roadway Name	Start	End
5	Anderson St	3	4
7	Anderson St	4	5
8	Anderson St	1	3
4	Marion Pugh dr	2	3
6	Marion Pugh dr	1	2
2	Texas Ave	1	2
1	University dr	1	2
3	University dr	2	3

Database sorted by Roadway name, start, and end			
ID	Roadway Name	Start	End
8	Anderson St	1	3
5	Anderson St	3	4
7	Anderson St	4	5
6	Marion Pugh dr	1	2
4	Marion Pugh dr	2	3
2	Texas Ave	1	2
1	University dr	1	2
3	University dr	2	3

Figure 5-8: The process of sorting data

Table 5-1: Grouping Process

ID	Roadway Name	Surface Type	Start	End	Group Number
8	Anderson St	Asphalt	1	3	1
5	Anderson St	Asphalt	3	4	1
7	Anderson St	Asphalt	4	5	1
9	Anderson St	Asphalt	8	9	2
6	Marion Pugh Dr	Asphalt	1	2	3
4	Marion Pugh Dr	Asphalt	2	3	3
2	Texas Ave	Asphalt	1	2	4
1	University Dr	Asphalt	1	2	5
10	University Dr	Asphalt	2	3	5
11	University Dr	Asphalt	4	5	6
3	University Dr	Asphalt	5	6	6

To find candidate segments for M&R treatment, a PCI threshold should be defined by the user to determine sections with lower PCI value than the threshold. Therefore, each group is assessed separately to find sections with unsatisfied PCI values; and then segments are formed out of groups with problematic sections.

In the next step, the total length of each segment was calculated by summing the lengths of all sections within that segment. Average PCI value of each segment was also calculated using Equation 5-3.

$$PCI_T = \frac{(PCI_1 \times L_1) + (PCI_2 \times L_2) + \cdots + (PCI_n \times L_n)}{L_1 + L_2 + \cdots + L_n} \qquad \textit{Equation 5-3}$$

Where PCI_T = Average segment PCI; PCI_1= PCI of section 1; L_1 = Length of section 1; n= number of sections within a segment.

5.9 Case Study

In this study, the city of College Station's road network is used to implement the new segmentation method. A Shapefile with 3877 sections was created using all the available data provide by the municipality of College Station. The initial map including all the sections can be seen in Figure 5-9. Each section contains information such as PCI, AADT, section length, number of lanes, roadway name, functional classification, speed limit, and surface type.

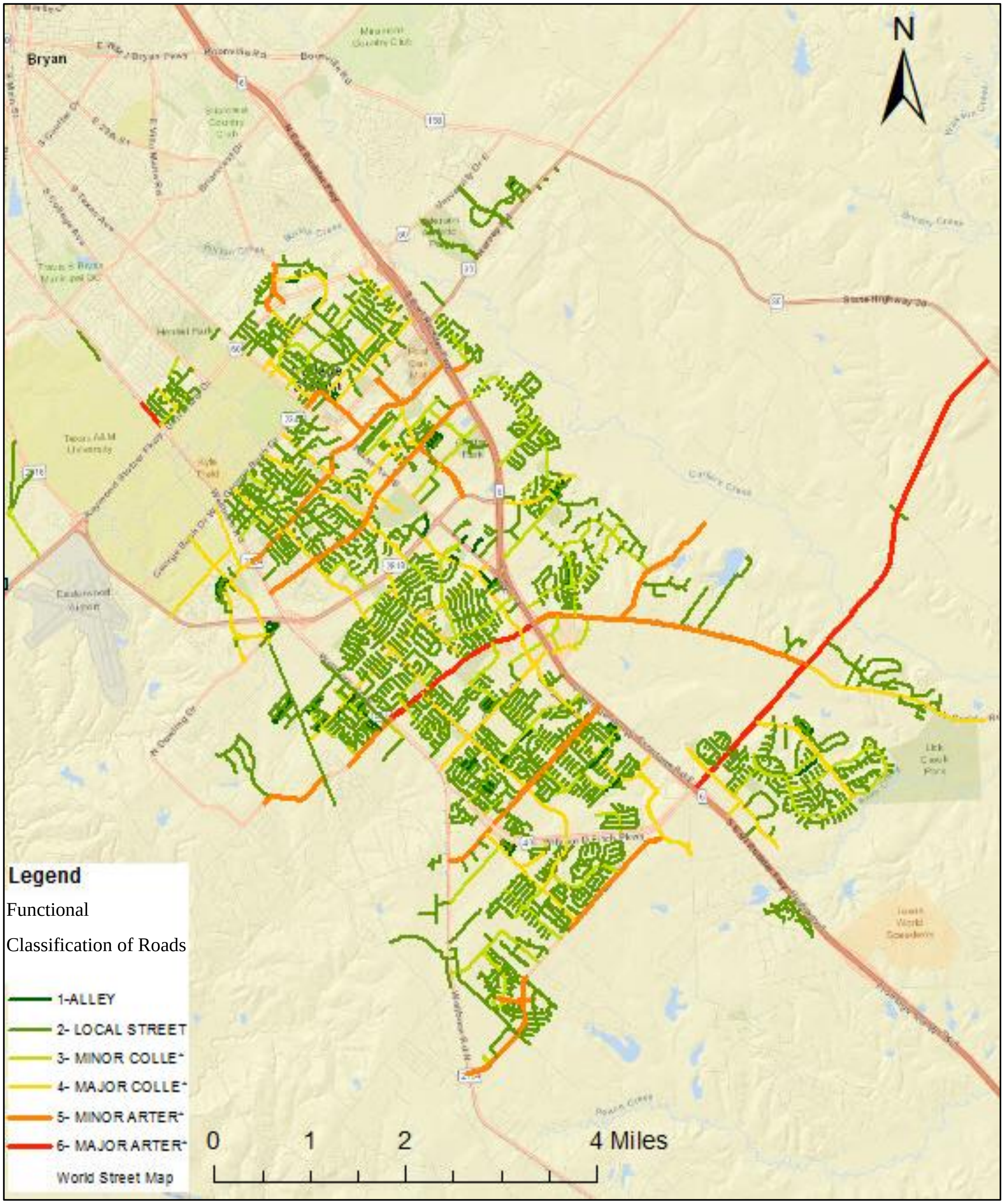
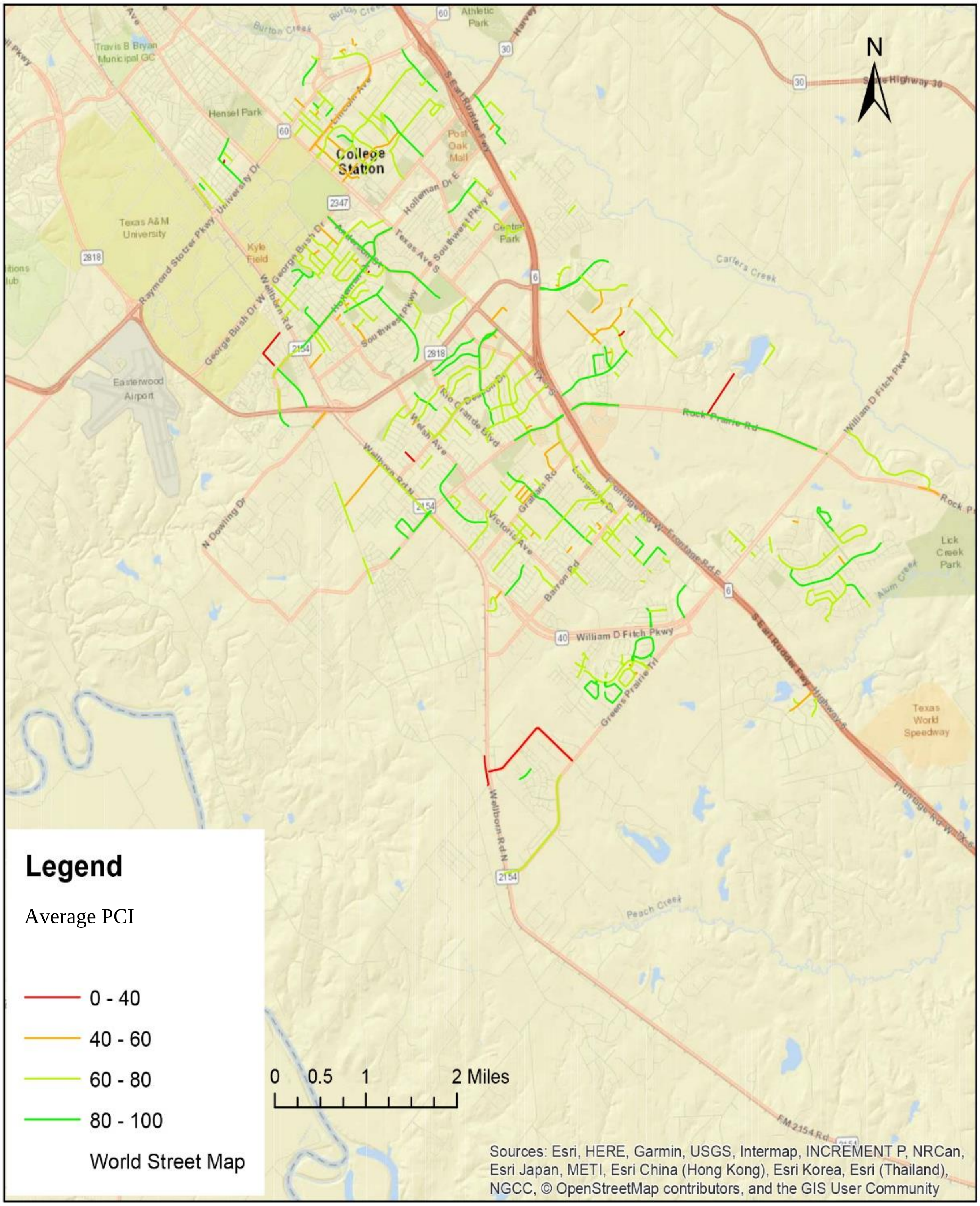


Figure 5-9: Map of the road network of College Station

Two scenarios were considered: one with PCI threshold of 70, and the other with PCI threshold of 80. For each scenario, the segmentation method was performed and the result of segmentation for each scenario is shown in Figure 5-10 and Figure 5-11. For the scenario with PCI threshold of 70, 261 segments were identified and for the scenario with PCI threshold of 80, 515 segments were created. As it can be seen, when the value of threshold increases, the number of segments also increases. This is because the number of sections with PCI value below 80 is more than the number of sections with PCI value below 70. To have a better sense on the effect of changing the threshold on segmentation result, the same process was repeated with PCI threshold of 60 and 90 and the result is shown in Table 5-2. As it can be seen, the number and total length of segments increases as the threshold increases. There is no meaningful change in the minimum and maximum length of each segment for different PCI thresholds, however, average segment length decreases with an increase in PCI threshold. At the end, average PCI distribution of segments and distribution of segments length are shown respectively in Figure 5-12, and Figure 5-13. As it can be seen, an increase in PCI threshold can cause average PCI distribution to shift to the right as more sections with high PCI values are being considered in the segmentation.

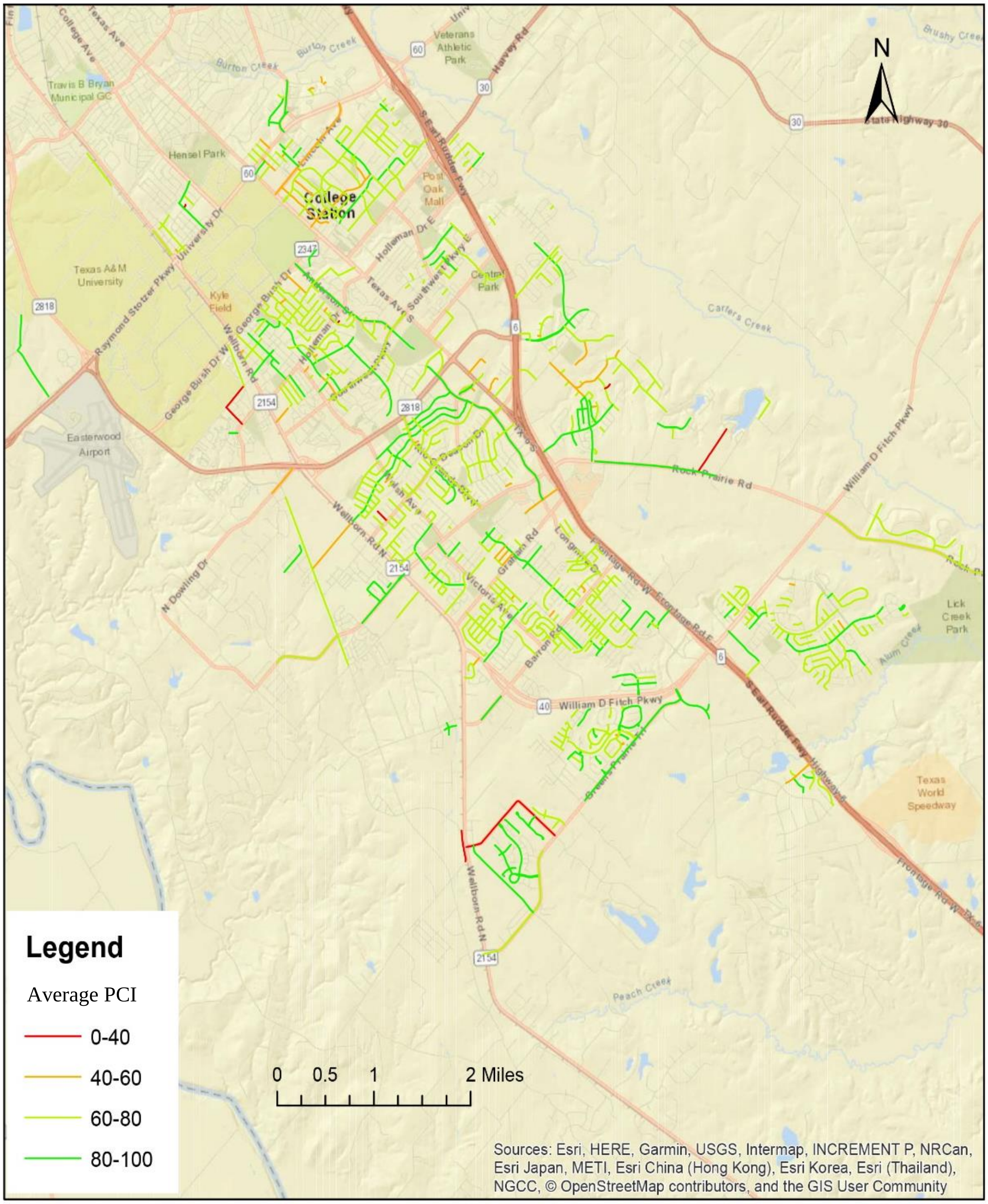
Table 5-2: Result of different segmentation threshold

PCI Threshold	60	70	80	90
Number of Segments	133	261	515	762
Minimum Segment Length (Mile)	0.021	0.021	0.021	0.015
Maximum Segment Length (Mile)	2.008	2.008	2.008	2.008
Average Segment Length (Mile)	0.404	0.332	0.271	0.251
Total Length of Segments (Mile)	53.814	86.707	139.41	191.167



Sources: Esri, HERE, Garmin, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), NGCC, © OpenStreetMap contributors, and the GIS User Community

Figure 5-10: Network segments with PCI threshold of 70



Sources: Esri, HERE, Garmin, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), NGCC, © OpenStreetMap contributors, and the GIS User Community

Figure 5-11: Network segments with PCI threshold of 80

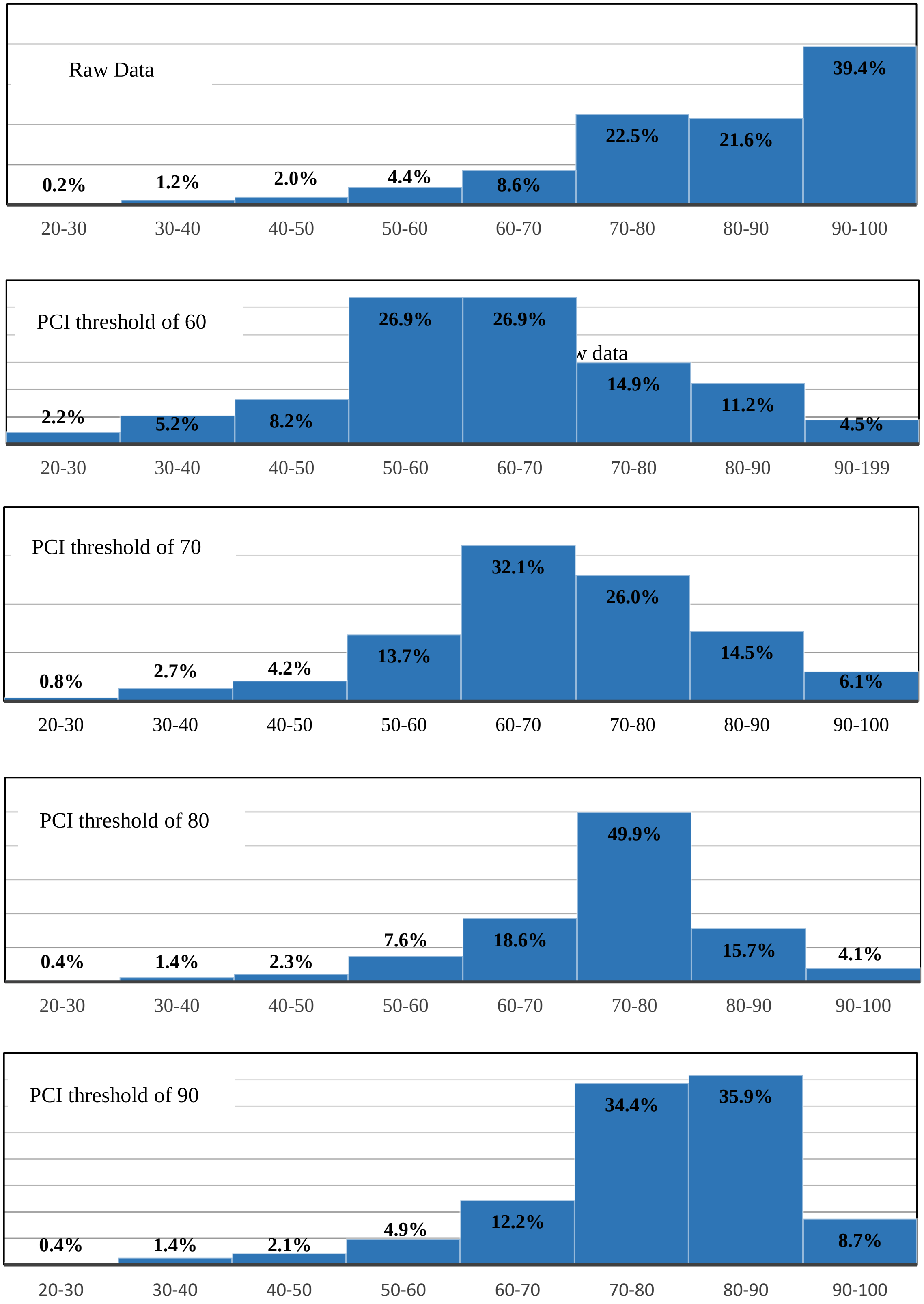


Figure 5-12: PCI distribution for raw data and various PCI thresholds

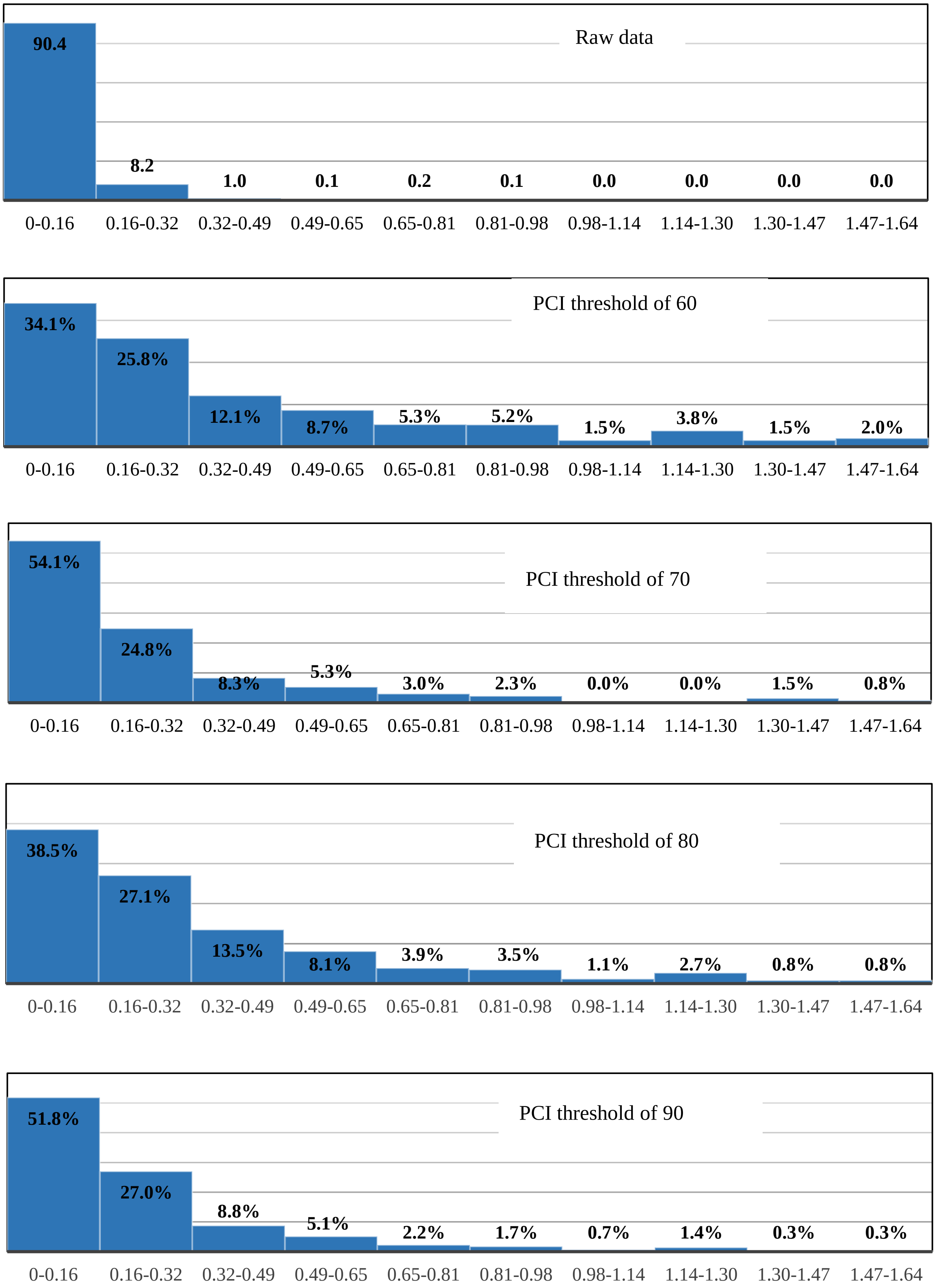


Figure 5-13: Segment length distribution for raw data and various PCI thresholds

5.10 Summary

In this chapter, the PDA method was used to perform segmentation for urban road network. The method was modified to make it applicable to urban areas which include a large number nodes and small arcs. Some of the problems encountered while running the segmentation algorithm in ArcGIS include preserving continuity and defining a unique representor for each node. A segmentation code which originally was developed for interurban road network in PHP was modified and transferred to Python to be compatible with ArcGIS software.

Four segmentation scenarios with different pavement condition threshold values were performed. According to the results, an increase in threshold value results in more number of segments, and smaller average segment length. However, minimum and maximum length of segments were not affected by threshold values.

6 SUMMARY OF RESEARCH EFFORTS, CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary and Conclusions

In this research, three problems associated with developing a Pavement Management System (PMS) for small to mid-size cities were discussed, and a solution was suggested for each of these problems. In chapter three, the comparison of condition of the road network based on PCI and IRI indicates that using IRI as a pavement condition indicator in urban areas can lead to misleading information about the condition of the road network. Additionally, the relationship between IRI and PCI was assessed. It was demonstrated that there is a weak correlation between IRI and PCI which is not enough to estimate one from another accurately. The lack of a meaningful relationship between these two pavement performance indicators in urban areas can be explained by distortions of IRI measurements in local road networks. Finally, it can be concluded that IRI may not be a suitable measure of pavement condition for urban streets due to external distortions. If local agencies allocate their resources to improve the methods they use to measure PCI or other condition indices which are originally developed for urban areas, they can get better and more accurate results.

In chapter four of the thesis, the problem with developing pavement prediction models without having a historical condition database was raised and a solution was suggested to overcome this problem. It was shown that, by having a rough estimate of the overall change in condition of the road network during one year, an iterative process can be used to estimate pavement prediction models coefficient. The methodology was tested using data from the city of College Station and one prediction model was developed for each M&R treatment.

In chapter five, problems associated with segmentation of the road network in urban areas were discussed in more detail. An automated segmentation method based on PDA approach was developed in Python. The segmentation code is compatible with ArcGIS which provides the user with all the visualization and analytical tools in ArcGIS. This segmentation method was successfully implemented for the city of College Station's road network considering four different thresholds. The results were presented in this chapter as well.

6.2 Recommendations

This research provides solutions for some common obstacles for developing PMS for small to mid-size cities. There are more advanced analyses that can be performed which couldn't be addressed in the scope of this thesis. Based on what has been done, the following recommendations are provided for future works:

- Ridge regression models can be used to develop more advanced PCI prediction models which consider other factors such as the width of the road, speed limit, AADT, and etc. in addition to IRI to reach more accurate models.
- A questionnaire can be developed for municipalities' engineers to reach a more precise estimate of PCI change during one year.
- A Python script can be written so that the whole process of segmentation can be done in ArcGIS without having to update the code after each analysis.
- A Python script can be developed so that the whole PMS procedure can be done in ArcGIS, not just the segmentation process.
- Guidelines could be developed based on the work completed in this study to assist small and mid-size cities to develop and implement PMS.

REFERENCES

Abudinen, Daniel, Luis G. Fuentes, and Juan S. Carvajal Muñoz. "Travel Quality Assessment of Urban Roads Based on International Roughness Index: Case Study in Colombia." Transportation Research Record: Journal of the Transportation Research Board 2612 (2017): 1-10.

Al-Omari, Bashar, and Michael I. Darter. "Effect of pavement deterioration types on IRI and rehabilitation." Transportation research record 1505 (1995).

American Association of State Highway and Transportation Officials (AASHTO). Pavement Management Guide, Second Edition 2012. American Association of State Highway and Transportation Officials. Washington, D.C. (2012).

Arhin, Stephen A., et al. "Predicting pavement condition index using international roughness index in a dense urban area." Journal of Civil Engineering Research 5.1 (2015): 10-17.

ASTM D 6433-07, Standard practice for roads and parking lots pavement condition index survey, West Conshohocken, PA. (2011).

Bennett, Christopher R. "Sectioning of road data for pavement management." 6th International Conference on Managing Pavements: The Lessons, The Challenges, The Way ahead. (2004).

Cadkin, Jennifer. "Understanding dynamic segmentation." ArcUser October-December (2002): 40-43.

Cafiso, Salvatore, and Alessandro Di Graziano. "Definition of homogenous sections in road pavement measurements." Procedia-Social and Behavioral Sciences 53 (2012): 1069-1079.

City of Nashville. (n.d). Metro Nashville Long Range Paving Plan, from http://mpw.nashville.gov/IMS/Paving/Documents/Chapter_2.pdf. Retrieved February 09, (2018).

Chiou, Chyi-Rong, Wei-Lun Tsai, and Yu-Fai Leung. "A GIS-dynamic segmentation approach to planning travel routes on forest trail networks in Central Taiwan." Landscape and urban Planning 97.4 (2010): 221-228.

Cottrell, Wayne D., et al. "Transportation infrastructure maintenance management: case study of a small urban city." Journal of Infrastructure Systems 15.2 (2009): 120-132.

Cuhadar, A., K. Shalaby, and S. Tasdoken. "Automatic segmentation of pavement condition data using wavelet transform." Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on. Vol. 2. IEEE, (2002).

Dewan, Shameem A., and Roger E. Smith. "Valuing pavement network assets and use of values as decision supports." Journal of infrastructure systems 11.4 (2005): 202-210.

Dewan, Shameem, and Roger Smith. "Estimating International Roughness Index from pavement distresses to calculate vehicle operating costs for the San Francisco Bay area." Transportation Research Record: Journal of the Transportation Research Board1816 (2002): 65-72.

Dueker, Kenneth J., and Ric Vrana. "Dynamic segmentation revisited: a milepoint linear data model." Journal of the Urban and Regional Information Systems Association 4.2 (1992): 94-105.

Elhadi, Hussain Mohammed Ahmed. "GIS, a tool for pavement management." (2009).

El Gendy, Amin, and Ahmed Shalaby. "Using quality control charts to segment road surface condition data." Proceedings of the Seventh International Conference on Managing Pavement Assets, Calgary, Alta. (2008).

Gagarine, Nicolas, James R. Mekemson, and Charles J. Nemmers. Assessing IRI vs. PI as a Measurement of Pavement Smoothness. No. OR06-17. (2006).

George, K. P., A. S. Rajagopal, and L. K. Lim. "Models for predicting pavement deterioration." Transportation Research Record 1215 (1989).

Hajek, Jerry, Thomas Kazmierowski, and Graham Musgrove. "Switching to international roughness index." Transportation Research Record: Journal of the Transportation Research Board 1643 (1998): 116-124.

Harrison, Frances, and Hyun-A. Park. "Comparative Performance Measurement: Pavement Smoothness." (2008).

Hauke, Jan, and Tomasz Kossowski. "Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data." Quaestiones geographical 30.2 (2011): 87-93.

Incorporated Places and Minor Civil Divisions Datasets: Subcounty Resident Population Estimates: April 1, 2010 to July 1, 2012: Texas". U.S. Census Bureau. Archived from the original on July 12, 2014. Retrieved April 21, (2014).

Kennedy, James, Ahmed Shalaby, and R. V. Cauwenberghe. "Dynamic segmentation of pavement surface condition data." 3rd Transportation Specialty Conference of the Canadian Society for Civil Engineering. (2000).

Loprencipe, Giuseppe, Antonio Pantuso, and Paola Di Mascio. "Sustainable pavement management system in urban areas considering the vehicle operating costs." Sustainability 9.3 (2017): 453.

Lovie, Pat. "Coefficient of variation." Encyclopedia of statistics in behavioral science (2005).

Misra, Rahul, and Animesh Das. "Identification of homogeneous sections from road data." International Journal of Pavement Engineering 4.4 (2003): 229-233.

Narciso, Paul John Ross. Development of A pavement maintenance and rehabilitation project formation and prioritization methodology that reflects agency priorities and improves network condition. Diss. (2013).

New York State Department of Transportation. Pavement Condition Assessment. Retrieved from https://www.dot.ny.gov/divisions/engineering/technical-services/technical-services-repository/pavement/nlp_cond_assess_manual.pdf. (2010).

Nunez, Maria Margarita, and Mohamed Y. Shahin. "Pavement condition data analysis and modeling." Transportation Research Record 1070 (1986): 125-132.

Nyerges, T. L. "Locational referencing and highway segmentation in a geographic information system." ITE journal60.3 (1990): 27-31.

Osorio, Alelí, et al. "Calibration and validation of condition indicator for managing urban pavement networks." Transportation Research Record 2455.1 (2014): 28-36.

Park, Kyungwon, Natacha E. Thomas, and K. Wayne Lee. "Applicability of the international roughness index as a predictor of asphalt pavement condition." Journal of Transportation Engineering 133.12 (2007): 706-709.

Rajagopal, Arudi. Developing pavement performance prediction models and decision trees for the city of Cincinnati. No. FHWA/OH-2006/14. (2006).

Reggin, Alan, et al. "Urban considerations for using road roughness to manage road networks." 7th International Conference on Managing Pavement Assets. (2008).

Rydholm, Timothy C., and David R. Luhr. "Visualizing pavement management data at the project level." 9th international conference on managing pavement assets. (2015).

Tavakoli, Amir, Mitchell S. Lapin, and J. Ludwig Figueroa. "PMSC: Pavement management system for small communities." Journal of transportation engineering 118.2 (1992): 270-280.

Thomas, Fridtjof. "Statistical approach to road segmentation." Journal of transportation engineering 129.3 (2003): 300-308.

Thomas, Fridtjof. "Automated road segmentation using a Bayesian algorithm." Journal of transportation engineering131.8: 591-598. (2005).

Transportation Officials. AASHTO guide for design of pavement structures, Vol. 1. AASHTO, (1993).

www.fhwa.dot.gov,www.fhwa.dot.gov/processes/statewide/related/highway_functional_classifications/fcauab.pdf, (2013).

Wu, Kan. Development of PCI-based Pavement Performance Model for Management of Road Infrastructure System. Arizona State University, (2015).

Xu, Guangyang, et al. "Pavement Deterioration Prediction Model and Project Selection for Kentucky Highways." 9th International Conference on Managing Pavement Assets. (2015).

APPENDIX A

```python
import arcpy
import os
import numpy as np
from operator import itemgetter


class Segmentation:
    _cs_threshold = None
    _cs_threshold = None                # threshold value for condition score
    _ds_threshold = None                # threshold value for distress score
    _mnr_trigger = None
    _mnr_trigger_param = None
    _user_id = None
    _esal_threshold = None
    _min_seg_len = None             # minimum segment length
    _max_seg_len = None             # maximum segment length
    _zvalue = None
    _sections = None                    # sections of base year in pmis
    _cs_based = None                    # condition score based segmentation result
    _ds_based = None                    # distress score based segmentation result
    _esal_based = None
    _groups = []                    # holds for group of roads divided by highway road bed, pavement type, and continuity
    _segments = []
    _group = None


    def __init__(self):
        #global database
        #global session
        self.base_year = "2017"
        self.table = "test"
        self.sections = []

        # $query = "SELECT ID, CURRENT_18KIP_MEAS, NUMBER_THRU_LANES, PRIOR_TREATMENT, YEAR_OF_PRIOR_TREATMENT, FISCAL_YEAR, RATING_CYCLE_CODE, SIGNED_HIG, ADJ_DISTRE,
        # ADJ_CONDIT, ADJ_RIDE_SCORE, SECT_LENGTH, START, BEG_REF_MARKER_DISP, END, END_REF_MARKER_DISP, PVMNT_TYPE_1,
        # AADT, TRUCK_AADT_PCT, DISTRICT_NAME, ZONE_NUMBER, SPEED_LIMIT_MAX, RATE_OF_DETERIORATION
        # FROM $table WHERE FISCAL_YEAR=$base_year AND RATING_CYCLE_CODE='P' AND PVMNT_TYPE_BROAD_CODE = 'A';

        fields = ["ID", "CURRENT_18", "NUMBER_THR", "PRIOR_TREA", "YEAR_OF_PR", "FISCAL_Y_1", "RATING_CYC",
                  "SIGNED_HIG", "ADJ_DISTRE", "ADJ_CONDIT", "ADJ_RIDE_S", "SECT_LENGT", "START", "BEG_REF__1",
                  "END", "END_REF__1", "PVMNT_TY_1", "AADT", "TRUCK_AADT", "DISTRICT_N", "ZONE_NUMBE", "SPEED_LIMI",
                  "RATE_OF_DE", "PVMNT_TYPE"]
        cursor = arcpy.da.SearchCursor("C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\python.shp",
                                  field_names=fields,
                                  where_clause="PVMNT_TYPE='A' AND RATING_CYC='P' AND FISCAL_Y_1=2017")
                                  #where_clause="PVMNT_TYPE='A' AND FISCAL_Y_1='2017' AND RATING_CYC='P'")

        for row in cursor:
            self.sections.append({"ID": row[0], "CURRENT_18": row[1], "NUMBER_THR": row[2], "PRIOR_TREA": row[3],
                                 "YEAR_OF_PR": row[4], "FISCAL_Y_1": row[5], "RATING_CYC": row[6], "SIGNED_HIG": row[7],
                                 "ADJ_DISTRE": row[8], "ADJ_CONDIT": row[9], "ADJ_RIDE_S": row[10], "SECT_LENGTH": row[11],
                                 "START": row[12], "BEG_REF__1": row[13], "END": row[14], "END_REF__1:": row[15],
                                 "PVMNT_TY_1": row[16], "AADT": row[17], "TRUCK_AADT": row[18], "DISTRICT_N": row[19],
                                 "ZONE_NUMBE": row[20], "SPEED_LIMI": row[21], "RATE_OF_DE": row[22]})
        self.grouping_tier2()
        self.sections = sorted(self.sections, key=itemgetter('SIGNED_HIG', 'START'))
        #for section in self.sections:
                #print section


    def _create_groups(self, similar_sections):

        if len(similar_sections) == 0:
            return

        while len(similar_sections) > 0:
            found_sections_for_group = True
            boundary_set = []       # Set of boundary points of all sections currently present in the group
            group = []

            # Push the first yet ungrouped section into its own group
            group.append(similar_sections[0])
            boundary_set.append(similar_sections[0]['START'])
            boundary_set.append(similar_sections[0]['END'])
            del similar_sections[0]

            while found_sections_for_group:
                found_sections_for_group = False

                for idx, cur_section in enumerate(similar_sections):

                    # Check if current section's boundary matches with boundary of any of the sections already present in group
                    if (cur_section['START'] in boundary_set) or (cur_section['END'] in boundary_set):
                        # merge section in group, push boundary points to boundary set

                        boundary_set.append(cur_section['START'])
                        boundary_set.append(cur_section['END'])
                        group.append(cur_section)
                        found_sections_for_group = True

                        # remove section from list
                        del similar_sections[idx]


                # Re-index similar sections array
                # similar_sections = similar_sections.values();

                # No more items to be added to current group, break and start new group
                # Push this group into list of groups
                if not found_sections_for_group:
                        # group = group.values()

                        # Sort sections in group by beg_ref_nbr and then end_ref_nbr to break ties
                        # array_multisort(array_column($group, 'ID'), SORT_ASC, $group);
                        # group = sorted(group, key=lambda k: k['ID'])

                        group = sorted(group, key=itemgetter('ID'))
                        #if len(group) > 1:
                            #print "After Sort Group - ", len(group), ": \n", group

                        cumLength = 0
                        # Assign CUM_LENGTH fields for all sections in the group
                        for group_section in group:
                            group_section['CUM_LENGTH_BEG'] = cumLength
                            group_section['CUM_LENGTH_END'] = cumLength + group_section['SECT_LENGTH']

                            cumLength += group_section['SECT_LENGTH']

                        self.groups.append(
                            {"GROUP": group, "TOTAL_LENGTH_END": cumLength, "TOTAL_LENGTH_BEG": cumLength - group[len(group) - 1]['SECT_LENGTH']})
                        group = []
                        # unset($group);
                        break


    # New method of grouping in tier-2 when the sections are not all in single line one after the other.
    # This works even when the Beginning is less than the ending unlike the assumption made in grouping in tier 1
    # and handles the Y formations and loops of sections.
    # Logic: Extract the similar sections (sections with same road ID and  Life code) and call the method
    # to create groups.
    def grouping_tier2(self):
        # get sections
        sections = self.sections

        # if the number of sections is zero, we are done
        if len(sections) == 0:
            return

        # initialize some variables
        #print sections[0]
        highway = sections[0]['SIGNED_HIG']

        # insert first section to similar sections
        self.groups = []
        similar_sections = []
        similar_sections.append(sections[0])

        for index, section in enumerate(sections, 1):
            # determine the section belongs to the same group
            # criteria are the signed highway roadbed id, continutiy and pavement family
            if ((section['SIGNED_HIG'] == highway)
                and (self.same_pvmnt_family(sections[index-1]['PVMNT_TY_1'], section['PVMNT_TY_1']))):
                # uncomment this line if offset is considered for continuity
                # $sections[$i]['BEG_REF_MARKER_DISP'] == $sections[$i-1]['END_REF_MARKER_DISP'] &&

                    # insert this section to current list of similar sections
                    similar_sections.append(section)

            else:
                # Create groups from the sections in this list
                self._create_groups(similar_sections)

                # clear list of similar sections
                # del similar_sections
                similar_sections = []

                # update some variables
                highway = section['SIGNED_HIG']

                # insert the first section to the similar sections
                similar_sections.append(section)

        # Create groups from the sections in this list
        self._create_groups(similar_sections)

        del similar_sections


    def do_segmentation_method2(self, sections, mnr_trigger, mnr_trigger_param, min_seg_len, max_seg_len):
        global session
        global database
        # xdebug_break();
        # $base_year = $_SESSION['baseyear'];

        # initialize parameters
        self.mnr_trigger = mnr_trigger
        self.mnr_trigger_param = mnr_trigger_param
        # xdebug_break();

        self.min_seg_len = min_seg_len
        if sections is not None:
            self.sections = sections
            ##### table = $_SESSION['pmistable'];
            ##### self.esal_threshold = $database->Query("SELECT avg(current_18kip_meas) as esal_threshold FROM $table where fiscal_year=$base_year and rating_cycle_code='P' and pvmnt_type_broad_code='A'");
            ##### self.esal_threshold = self.esal_threshold[0]['esal_threshold']

            # Tier 2: changed to new grouping (grouping_tier2)
            # $this->grouping();
            self.grouping_tier2()

        groups = self.groups
        #for group in groups:
            #print "Another Groupppp: ", group
            #print "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^"
```

```
                segment_id = 1
                segments = []
                for i, groups_i in enumerate(groups):
                    # if ($i==0 || $i==579)
                    #   xdebug_break();
                    group = groups_i['GROUP']
                    segment = []
                    boundaries = []
                    # xdebug_break();

                    # Mark beginning and ending (boundaries) for sections that are members of current group and need MNR
                    # based on condition score
                    # boundaries contains the sections that form the boundary of prospective segments from the current group
                    for j, group_j in enumerate(group):
                        # if ($group[$j]['SIGNED_HIG'] == 'FM2620 K' && $group[$j]['START'] == 404
                        #   && $group[$j]['BEG_REF_MARKER_DISP'] == 1.5)
                        #   xdebug_break();
                        if ((mnr_trigger_param == "Condition Score" and group_j['ADJ_CONDIT']<mnr_trigger) or
                            (mnr_trigger_param == "Distress Score" and group_j['ADJ_DISTRE']>mnr_trigger)):
                            group_j['NEED_MNR'] = True
                            if not segment:
                                group_j['IS_BEG'] = True
                                boundaries.append({'INFO': group_j, 'INDEX': j})

                            segment.append(j)

                        else:
                            group_j['NEED_MNR'] = False
                            if segment:
                                last = segment[len(segment)-1]
                                group[last]['IS_END'] = True
                                #if (group[last]['IS_BEG'] is not None) and (group[last]['IS_BEG']):
                                if ('IS_BEG' in group[last]) and (group[last]['IS_BEG']):
                                    boundaries.pop()

                                boundaries.append({'INFO': group[last], 'INDEX': last})

                                #del segment
                                segment = []

                    if segment:
                        last = segment[len(segment)-1]
                        group[last]['IS_END'] = True
                        #if (group[last]['IS_BEG'] is not None) and (group[last]['IS_BEG']):
                        if ('IS_BEG' in group[last]) and (group[last]['IS_BEG']):
                            boundaries.pop()

                        boundaries.append({'INFO': group[last], 'INDEX': last})
                        #unset($segment);
                        segment = []


                    '''''
```

```
         * extent 1
         '''
        # For boundaries of prospective segments that are separated by less than extent1 distance,
        # edit and remove boundaries so that only 1 prospective segments' boundaries remain.
        result_boundaries = boundaries
        extent1 = min_seg_len   # + 0.6;    5/1 Change
        for j, boundary_j in enumerate(boundaries):
            # if ($group[$j]['SIGNED_HIG'] == 'FM2620 K' && $group[$j]['START'] == 404
            #   && $group[$j]['BEG_REF_MARKER_DISP'] == 1.5)
            #   xdebug_break();
            index = boundary_j['INDEX']
            boundary = boundary_j['INFO']

            # TODO: This logic is probably wrong.
            # TODO: Since the latter if condition already unsets the IS_END flag of the previous section
            # TODO: the current section will never find a previous ending boundary with IS_END flag within extent1.
            # TODO: Ideally we should unset both in the same iteration and there should be only 1 if condition.
            #if (boundary['IS_BEG'] is not None) and boundary['IS_BEG']:
            if ('IS_BEG' in boundary) and boundary['IS_BEG']:
                if j - 1 >= 0:
                    prev_boundary = boundary_j['INFO']
                    if boundary['CUM_LENGTH_BEG'] - prev_boundary['CUM_LENGTH_END'] >= extent1:
                        pass
                        # if ($boundary['CUM_LENGTH_BEG']-$prev_boundary['CUM_LENGTH_BEG']-$extent1 >= 0.0000001)
                    else:
                        if ('IS_END' in prev_boundary) and prev_boundary['IS_END']:
                            # there is End rear then make this beg cancel
                            # unset($result_boundaries[$j]['INFO']['IS_BEG']);
                            del result_boundaries[j]['INFO']['IS_BEG']

            #if (boundary['IS_END'] is not None) and boundary['IS_END']:
            if ('IS_END' in boundary) and boundary['IS_END']:
                if j + 1 <= len(boundaries) - 1:
                    next_boundary = boundaries[j+1]['INFO']
                    if next_boundary['CUM_LENGTH_BEG'] - boundary['CUM_LENGTH_END'] >= extent1:
                        pass
                        # if ($next_boundary['CUM_LENGTH_END']-$boundary['CUM_LENGTH_END']-$extent1 >= 0.0000001) {
                    else:
                        if next_boundary['IS_BEG']:
                            # there is Beg front then make this end cancel
                            # unset($result_boundaries[$j]['INFO']['IS_END']);
                            del result_boundaries[j]['INFO']['IS_END']


            #is_beg_set = result_boundaries[j]['INFO']['IS_BEG'] is not None
            #is_end_set = result_boundaries[j]['INFO']['IS_END'] is not None
```

```
                        is_beg_set = 'IS_BEG' in result_boundaries[j]['INFO']
                        is_end_set = 'IS_END' in result_boundaries[j]['INFO']
                        if is_beg_set and is_end_set and result_boundaries[j]['INFO']['IS_BEG'] and result_boundaries[j]['INFO']['IS_END']:
                            result_boundaries[j]['INFO']['MARK'] = "B&E"
                        elif is_beg_set and result_boundaries[j]['INFO']['IS_BEG']:
                            result_boundaries[j]['INFO']['MARK'] = "B"
                        elif is_end_set and result_boundaries[j]['INFO']['IS_END']:
                            result_boundaries[j]['INFO']['MARK'] = "E"

                    boundaries = result_boundaries
                    result_boundaries = []
                    #del result_boundaries


                    # if (isset($sections) && $boundaries[0]['INFO']['SIGNED_HIG'] == "FM1452 K") {
                    # for ($j=0 ; $j<count($boundaries) ; $j++) {
                    #   if (isset($boundaries[$j]['INFO']['MARK'])) {
                    #       print $boundaries[$j]['INFO']['MARK'].",";
                    #       $this->print_road($group, $boundaries[$j]['INDEX']);
                    #   }
                    # }
                    # xdebug_break();
                    # }


                    '''''
                     * extent2
                    '''
                    # xdebug_break();
                    result_boundaries = boundaries
                    extent2a = extent1 + min_seg_len/2
                    extent2b = extent1 + min_seg_len
                    for j, boundary_j in enumerate(boundaries):
                        # if ($group[$j]['SIGNED_HIG'] == 'FM2620 K' && $group[$j]['START'] == 404
                        #   && $group[$j]['BEG_REF_MARKER_DISP'] == 1.5)
                        #   xdebug_break();
                        # if ($boundaries[$j]['INFO']['SIGNED_HIG'] == 'FM1696 K' && $boundaries[$j]['INFO']['START'] == 654
                        #       && $boundaries[$j]['INFO']['BEG_REF_MARKER_DISP'] == 1.5)
                        #       xdebug_break();

                        if ('MARK' in boundary_j['INFO']) and (boundary_j['INFO']['MARK'] == "B"):
                            # B&E rear
                            if j - 1 >= 0:
                                # if ($boundaries[$j]['INFO']['CUM_LENGTH_BEG']-$boundaries[$j-1]['INFO']['CUM_LENGTH_BEG']-$extent2a >= 0.0000001)
                                if boundary_j['INFO']['CUM_LENGTH_BEG'] - boundaries[j-1]['INFO']['CUM_LENGTH_END'] >= extent2a:
                                    pass
                                else:
                                    if boundaries[j-1]['INFO']['MARK'] == "B&E":
                                        # $result_boundaries[$j-1]['INFO']['MARK'] = "B";
                                        # unset($result_boundaries[$j]['INFO']['MARK']);

                                        del result_boundaries[j]['INFO']['MARK']


                        elif ('MARK' in boundary_j['INFO']) and (boundary_j['INFO']['MARK'] == "E"):
                            # B&E front
                            if j + 1 <= len(boundaries) - 1:
                                # if ($boundaries[$j+1]['INFO']['CUM_LENGTH_END']-$boundaries[$j]['INFO']['CUM_LENGTH_END']-$extent2a >= 0.0000001)
                                if (boundaries[j+1]['INFO']['CUM_LENGTH_BEG'] - boundary_j['INFO']['CUM_LENGTH_END'] >= extent2a):
                                    pass
                                else:
                                    if boundaries[j+1]['INFO']['MARK'] == "B&E":
                                        # $result_boundaries[$j+1]['INFO']['MARK'] = "E";
                                        # unset($result_boundaries[$j]['INFO']['MARK']);
                                        del result_boundaries[j]['INFO']['MARK']

                        elif ('MARK' in boundary_j['INFO']) and (boundary_j['INFO']['MARK'] == "B&E"):
                            # if ($boundaries[$j]['INFO']['SIGNED_HIG'] == 'FM2620 K' && $boundaries[$j]['INFO']['START'] == 0)
                            # xdebug_break();
                            no_e_in_rear = True
                            no_be_in_rear = True
                            no_b_in_front = True
                            no_be_in_front = True
                            if j -  1 >= 0:
                                # if ($boundaries[$j]['INFO']['CUM_LENGTH_BEG']-$boundaries[$j-1]['INFO']['CUM_LENGTH_BEG']-$extent2b >= 0.0000001)
                                if boundary_j['INFO']['CUM_LENGTH_BEG'] - boundaries[j-1]['INFO']['CUM_LENGTH_END'] >= extent2b:
                                    pass
                                else:
                                    # count B&E and E in rear
                                    if boundaries[j-1]['INFO']['MARK'] == "B&E":
                                        # $result_boundaries[$j-1]['INFO']['MARK'] = "B";
                                        # $result_boundaries[$j]['INFO']['MARK'] = "E";
                                        no_be_in_rear = False

                                    # if ($boundaries[$j]['INFO']['CUM_LENGTH_BEG']-$boundaries[$j-1]['INFO']['CUM_LENGTH_BEG']-$extent2a >= 0.0000001) {
                                    if boundary_j['INFO']['CUM_LENGTH_BEG'] - boundaries[j-1]['INFO']['CUM_LENGTH_END'] >= extent2a:
                                        pass

                                    elif boundaries[j-1]['INFO']['MARK'] == "E":
                                        # unset($result_boundaries[$j-1]['INFO']['MARK']);
                                        # $result_boundaries[$j]['INFO']['MARK'] = "E";
                                        no_e_in_rear = False

                            if j + 1 <= len(boundaries) - 1:
                                # if ($boundaries[$j+1]['INFO']['CUM_LENGTH_END']-$boundaries[$j]['INFO']['CUM_LENGTH_END']-$extent2b >= 0.0000001)
                                if boundaries[j+1]['INFO']['CUM_LENGTH_BEG'] - boundary_j['INFO']['CUM_LENGTH_END'] >= extent2b:
                                    pass
                                else:
                                    # count B&E and B in front
                                    if boundaries[j+1]['INFO']['MARK'] == "B&E":
                                        # $result_boundaries[$j+1]['INFO']['MARK'] = "E";
                                        # $result_boundaries[$j]['INFO']['MARK'] = "B";
                                        no_be_in_front = False

                                    # if ($boundaries[$j+1]['INFO']['CUM_LENGTH_END']-$boundaries[$j]['INFO']['CUM_LENGTH_END']-$extent2a >= 0.0000001) {
                                    if boundaries[j+1]['INFO']['CUM_LENGTH_BEG'] - boundary_j['INFO']['CUM_LENGTH_END'] >= extent2a:
                                        pass

                                    elif boundaries[j+1]['INFO']['MARK'] == "B":
                                        # unset($result_boundaries[$j+1]['INFO']['MARK']);
                                        # $result_boundaries[$j]['INFO']['MARK'] = "B";
                                        no_b_in_front = False


                            result1 = ""
                            result2 = ""
                            if no_e_in_rear: #&& $no_be_in_rear)
                                result1 = "B"
                            if no_b_in_front: #&& $no_be_in_front)
                                result2 = "E"
                            if result1 != "" and result2 != "":
                                result_boundaries[j]['INFO']['MARK'] = "B&E"
                            elif result1 != "":
                                result_boundaries[j]['INFO']['MARK'] = "B"
                            elif result2 != "":
                                result_boundaries[j]['INFO']['MARK'] = "E"
                            else:
                                #unset($result_boundaries[$j]['INFO']['MARK']);
                                del result_boundaries[j]['INFO']['MARK']


                        # update boundary info
                        if 'MARK' in result_boundaries[j]['INFO']:
                            group[result_boundaries[j]['INDEX']]['MARK'] = result_boundaries[j]['INFO']['MARK']

                        else:
                            group[result_boundaries[j]['INDEX']]['MARK'] = None


                    boundaries = result_boundaries
                    # xdebug_break();
                    '''''
                     * extent 2
                    '''

                    # for ($j=0 ; $j<count($boundaries) ; $j++) {
                    #   if (isset($boundaries[$j]['INFO']['MARK'])) {
                    #       print $boundaries[$j]['INFO']['MARK'].",";
                    #       $this->print_road($group, $boundaries[$j]['INDEX']);
                    #   }
```

```
        # }






        '''''
         * extent 3
        '''
        extent = min_seg_len / 2 + 0.1
        #unset($this->group);
        #$this->group = $group;
        self.group = group
        # xdebug_break();
        for j, boundary_j in enumerate(boundaries):
            # if ($group[$j]['SIGNED_HIG'] == 'FM2620 K' && $group[$j]['START'] == 404
            #   && $group[$j]['BEG_REF_MARKER_DISP'] == 1.5)
            #   xdebug_break();
            if 'MARK' in boundary_j['INFO']:
                if boundary_j['INFO']['MARK'] == "B":
                    self._extend_to_front("B", boundary_j['INDEX'], extent, False)
                elif boundary_j['INFO']['MARK'] == "E":
                    self._extend_to_rear("E", boundary_j['INDEX'], extent, False)
                elif boundary_j['INFO']['MARK'] == "B&E":
                    self._extend_to_rear("B&E", boundary_j['INDEX'], extent, False)
                    self._extend_to_front("B&E", boundary_j['INDEX'], extent, False)


        group = self.group

        group = self.partial_split(group, max_seg_len)

        # for ($j=0 ; $j<count($group) ; $j++) {
        #   if (isset($group[$j]['MARK'])) {
        #       print $group[$j]['MARK'].",";
        #       $this->print_road($group, $j);
        #   }
        # }
        # xdebug_break();

        # sections to segments
        highway = group[0]['SIGNED_HIG']
        segment = []
        try:
            prev_highway
        except NameError:
            segment_id = 0
        else:
            if highway != prev_highway:
                segment_id = 0
            #for j, group_j in enumerate(group):
            #print "Lengthhhhhhh Grouppppp: ", len(group)
            #print "\n"
            for j in range(len(group)):
                if ('MARK' in group[j]) and (group[j]['MARK'] == "B&E" or group[j]['MARK'] == "B"):
                    segment_id += 1
                    group[j]['SEGMENT_ID'] = highway + ' ' + str(segment_id)

                    # if (count($segments)==245)
                    # xdebug_break();
                    segment.append(group[j])
                    if group[j]['MARK'] == "B":
                        j += 1
                        while (j <= len(group)-1) and ((('MARK' in group[j]) and group[j]['MARK'] != "E") or 'MARK' not in group[j]):

                            group[j]['SEGMENT_ID'] = highway + ' ' + str(segment_id)

                            segment.append(group[j])
                            j += 1

                        if j == len(group):
                            j -= 1

                        group[j]['SEGMENT_ID'] = highway + ' ' + str(segment_id)

                        segment.append(group[j])

                    segments.append(segment)
                    #unset($segment);
                    segment = []


        prev_highway = highway
        self.groups[i]['GROUP'] = group

    #del self.segments
    #unset($this->segments);
    # xdebug_break();
    self.segments = segments

    # split long segments

    # xdebug_break();

    # if sections is None:
    #   self.create_group_and_table(2, True)
    # else:
    #   self.create_group_and_table(2, False)
    segments = self.segments
    # xdebug_break();
    return segments


def _extend_to_rear(self, mark, start, extent, force):
    group = self.group
    found = False
    group_length = 0
    # calculate group length
    if mark == "B&E":
        group_length = group[start]['SECT_LENGTH']
        beg = start
        found = True

    elif mark == "E":
        group_length = group[start]['SECT_LENGTH']
        i = start

        # find matched B
        i -= 1
        while i >= 0:
            group_length += group[i]['SECT_LENGTH']
            if ('MARK' in group[i]) and group[i]['MARK'] == "B":
                beg = i
                found = True
                break

            elif 'MARK' in group[i]:
                break

            if group[i]['NEED_MNR']:
                first = i

            i =- 1

        if not found:
            try:
                first
            except NameError:
                print "The variable found was not defined"
                group_length = group[start]['SECT_LENGTH']
                beg = start
                found = True
            else:
                beg = first
                group_length = 0
            # if ($first == null) {
            #   $group_length = $group[$start]['SECT_LENGTH'];
            #   $beg = $start;
            #   $found = true;
            # }
            # else {
            # }
            # xdebug_break();


    # mismatch found
    if not found:
        print "Error - found not defined";
        # temp B
        # xdebug_break();
        # $this->extend_to_front("B", $beg, $extent, true);
        # $group = $this->group;


    # Need Merge
    if group_length < self.min_seg_len or force:
        i = beg - 1
        # extend B if possible

        # go back
        while i >= 0:
            # look ahead
            j = i + 1
            cum_length = 0
            stop = False
            discovered = False
            while True:
                # $cum_length += $group[$j]['SECT_LENGTH'];
                # we found NM
                # if ($group[0]['SIGNED_HIG'] == "BS0021HK")
                #   xdebug_break();
                if group[j]['NEED_MNR'] and cum_length < extent:
                    # move end
                    if 'MARK' in group[beg] and group[beg]['MARK'] == "B&E":

                        group[beg]['MARK'] = "E"
                    elif 'MARK' in group[beg]:
                        del group[beg]['MARK']

                    beg = i
                    group[i]['MARK'] = "B"
                    discovered = True
                    break

                cum_length += group[j]['SECT_LENGTH']   # FROM LINE 516 TO 529
                if stop:
                    break
                j += 1
                if 'MARK' in group[j]:
                    stop = True

            if not discovered:
                break
            i -= 1

    self.group = group


def _extend_to_front(self, mark, start, extent, force):
    group = self.group
    # if ($group[0]['SIGNED_HIG'] == "FM1618 K")
    #   xdebug_break();
    found = False
    group_length = 0
    # calculate group length
    if mark == "B&E":
        group_length = group[start]['SECT_LENGTH']
        end = start
        found = True

    elif mark == "B":
        group_length = group[start]['SECT_LENGTH']
        i= start

        # find matched E
        i += 1
        while i <= len(group) - 1:
            group_length += group[i]['SECT_LENGTH']
            if ('MARK' in group[i]) and group[i]['MARK'] == "E":
                end = i
                found = True
                break

            elif 'MARK' in group[i]:
                break

            if group[i]['NEED_MNR']:
                last = i

            i += 1

        if not found:
            # if ($last==null) {
            #   $group_length = $group[$start]['SECT_LENGTH'];
            #   $end = $start;
            #   $found = true;
            # }
            # else {
            try:
                last
            except NameError:
                print "The variable found was not defined"
                group_length = group[start]['SECT_LENGTH']
                end = start
                found = True
            else:
                #print "sure, it was defined."
                end = last
                group_length = 0
            # }
            # xdebug_break();



    # mismatch found
    if not found:
        print "Error - found not defined"
        # temp B
        # xdebug_break();
        # $this->extend_to_rear("E", $end, $extent, true);
        # $group = $this->group;


    # Need Merge
    if group_length < self.min_seg_len or force:
        i = end + 1
        # extend E if possible

        # go front
        while i <= len(group) - 1:
            # look back
            j = i - 1
            cum_length = 0
            stop = False
            discovered = False
            while True:
                # $cum_length += $group[$j]['SECT_LENGTH'];
                # we found NM
                if group[j]['NEED_MNR'] and cum_length < extent:
                    # move end
                    if 'MARK' in group[end] and group[end]['MARK'] == "B&E":
                        group[end]['MARK'] = "B"

                    elif 'MARK' in group[end]:
```

```
                            del group[end]['MARK']

                        end = i
                        group[i]['MARK'] = "E"
                        discovered = True
                        break

                    cum_length += group[j]['SECT_LENGTH'] # FROM LINE 608 -> TO 622
                    if stop:
                        break
                    j -= 1
                    if 'MARK' in group[j]:
                        stop = True

                if not discovered:
                    break

                i += 1

        self.group = group


    def _get_last_key(array):
        return array.keys()[-1]
        #end(array);
        #return key(array)

    def same_pvmnt_family(self, pvmnt_code1, pvmnt_code2):
        # if they are the same pvmnt family
        if self.get_pvmnt_family(pvmnt_code1) == self.get_pvmnt_family(pvmnt_code2):
            return True
        # if not
        else:
            return False

    ''''' get pvmnt family based on pvmnt code
     * 4,5,9 => A
     * 7,8 => B
     * 6,10 => C
     '''
    def get_pvmnt_family(self, pvmnt_code):
        # manual override for pvmnt code 01
        if pvmnt_code == '01':
            pvmnt_code = '05'

        if pvmnt_code == '04' or pvmnt_code == '05' or pvmnt_code == '09':
            return 'A'
        elif pvmnt_code == '07' or pvmnt_code == '08':
            return 'B'
        else:
            return 'C'

    def partial_split(self, group ,max_seg_len):
        #for ($j=0 ; $j<count($group) ; $j++) {
        for j,group_j in enumerate(group):
            cum_length = 0
            beg_index = j
            if ('MARK' in group_j) and (group_j['MARK'] == "B&E" or group_j['MARK'] == "B"):
```

```
                        cum_length += group_j['SECT_LENGTH']
                        if group_j['MARK'] == "B":
                            j += 1
                            cum_length += group_j['SECT_LENGTH']
                            while j <= len(group)-1 and ('MARK' in group_j and group_j['MARK'] != "E") or ('MARK' not in group_j):
                                j += 1
                                cum_length += group_j['SECT_LENGTH']

                        end_index = j

                        # SPLIT
                        if cum_length > max_seg_len:
                            # xdebug_break();
                            start = group[beg_index]['CUM_LENGTH_BEG']
                            end = group[end_index]['CUM_LENGTH_BEG']
                            howmany = math.ceil((end-start)/max_seg_len)
                            count = end_index - beg_index + 1
                            num_subsections = math.ceil(count/howmany)
                            id = 1
                            count = 0
                            #for ($i=$beg_index ; $i<=$end_index ; $i++) {
                            for i in range(beg_index, end_index+1):
                                if (group[i]['CUM_LENGTH_END'] > start and
                                    group[i]['CUM_LENGTH_END'] <= end):
                                    if (id % num_subsections) == 0:
                                        group[i]['MARK'] = "E"
                                        group[i+1]['MARK'] = "B"
                                        count += 1

                                    if count == howmany-1:
                                        break
                                    id += 1

                return group


        segment = Segmentation()
        segments = segment.do_segmentation_method2(segment.sections, 80, "Condition Score", 2, 10)
        segment_values = []
        segment_ids = []
        arcpy.CopyFeatures_management("C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\python.shp","C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\CSSegment.shp")
        arcpy.AddField_management("C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\CSSegment.shp","seg_id","SHORT")
        arcpy.AddField_management("C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\CSSegment.shp","pci_len","DOUBLE")
        counter = 0
        for segment_instance in segments:
            counter += 1
            temp_segment_id = {'Segment': segment_instance, 'ids': []}
            for section_i in segment_instance:
                temp_segment_id['ids'].append(section_i['ID'])

            # if len(segment_instance[0].items()) == 30:
            #   segment_values.append(tuple(segment_instance[0].values()))

            segment_ids.append(temp_segment_id)
            #print temp_segment_id['Segment'][0]['SEGMENT_ID'], ": ", temp_segment_id['ids'], "\n"
            with arcpy.da.UpdateCursor("C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\CSSegment.shp",["seg_id", "ID" ,"ADJ_CONDIT", "SECT_LENGT", "pci_len"]) as cursor:
                for row in cursor:
                    if row[1] in temp_segment_id['ids']:
                        row[4] = row[2] * row[3]
                        row[0] = counter
                        cursor.updateRow(row)
                    else:
                        pass
                del cursor

        statistics = [['SECT_LENGT', 'SUM'], ['pci_len', 'SUM']]
        arcpy.Dissolve_management("C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\CSSegment.shp","C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\CSSegment_Final.shp", "seg_id", statistics)
        with arcpy.da.UpdateCursor("C:\\Sina\\Thesis\\GIS\\Latest Code\\corrected darabase\\80\\CSSegment_Final.shp",["SUM_SECT_L", "SUM_pci_le"]) as cursor:
            for row in cursor:
                row[1] = row[1] / row[0]
                cursor.updateRow(row)

            del cursor
```