

ABSTRACT

KADAM, PRAFUL SANJEEV. Design of a Public Logistics Network for the Local Distribution of Packages. (Under the direction Dr. Michael Kay.)

This research presents the design of a public logistics network (PLN) that covers the Raleigh-Durham-Cary combined statistical area of North Carolina (RDC). The design goal of the network is aimed at minimizing the delivery time of packages to customer locations while ensuring efficient operation of the network. The PLN will cover the distribution of all day-to-day commercial and domestic products. In RDC in order to achieve the finest possible resolution of population, the census block was used to locate population centroids because it was the lowest level geographic entity. Data was collected from multiple sources to estimate upper and lower bounds on daily per capita package demand in RDC. All the census blocks (demand points) were connected to the road network in RDC, followed by locating the DCs at intersection points and the design of the PLN. The average delivery time of a package within the generated PLN was calculated as the sum of the time required by a package reach to the DC closest to the customer location from the DC farther away and the local distribution time of a package from the closest DC to the customer location. Vehicle routes were designed to distribute the packages from the DC to the allocated customer locations (census blocks). Local distribution time constitutes the local transportation time and the waiting time of a package for an available delivery truck. A heuristic utilizing the alternate location-allocation and neighborhood search procedures was developed to find locations of the DCs in the PLN for which average delivery time is minimized. The effects of various values of daily per-capita package demand and package loading-unloading times on best DC locations and delivery times were studied.

Design of a Public Logistics Network for the Local Distribution of Packages

by
Praful Sanjeev Kadam

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Industrial Engineering

Raleigh, North Carolina

2011

APPROVED BY:

Donald Warsing

Christian Rossetti

Michael G. Kay
Chair of Advisory Committee

DEDICATION

To Aai and Baba

BIOGRAPHY

Praful Sanjeev Kadam was born on 26th March 1987 in Nanded in the state of Maharashtra, India. He has obtained Bachelor of Technology degree in Mechanical Engineering from College of Engineering, Pune. During his Master's program in Industrial Engineering at North Carolina State University, he concentrated on areas of Operations Research and Supply Chain Logistics.

ACKNOWLEDGMENTS

I would like to thank specially to Dr. Michael Kay for giving me this opportunity to work on this research. This research experience helped me gain profound knowledge of variety of areas in Supply Chain Logistics and Dr. Kay always supported and provided me unparalleled guidance throughout this journey. I would also like to thank my committee members Dr. Donald Warsing and Dr. Christian Rossetti for their guidance and feedback.

Finally, I am grateful to my parents, Mr. Sanjeev Kadam and Ms. Shubhangi Kadam, my lovely sisters and nieces for their blessings and love.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER 1: Introduction	1
1.1 Introduction	1
1.2 Objectives	4
1.3 Background and Overview	5
1.4 Literature Review	9
1.4.1 PLN Design	9
1.4.2 Design of Vehicle Schedule and Routes	10
1.5 Thesis Outline	13
CHAPTER 2: Data Collection	14
2.1 Introduction	14
2.2 Data Sources	14
2.3 Geographic Boundary Limits of RDC	16
2.4 Collection of Demographic Data for Different Levels of Geography	17
2.5 Per Capita Daily Package Demand in RDC	19
2.5.1 Package Deliveries by USPS	19
2.5.2 Package Demand Fulfilled by Customer Trips	21
2.6 Package Carrying Capacity of a Vehicle	26
CHAPTER 3: Design of the PLN	27
3.1 Introduction	27
3.2 Generation of the URN and the Network of Public DCs	27
3.3 Calculation for Average Package Delivery Time	30
3.4 Vehicle Routing from DCs to Customer Locations	35
3.4.1 Estimation of Approximate Vehicle Route Distance	35
3.4.2 Preprocessing	38

3.4.3 Estimation of Vehicle Route Distance for Census Block and Group of Census Blocks	43
3.5 Search Heuristics.....	49
3.5.1 Creating the Set of Potential Points.....	53
3.5.2 ALA Using Neighborhood Search	54
Chapter 4: Results, Conclusion and Future Work	60
4.1 Results	60
4.1.1 Best Case Scenario	60
4.1.2 Effect of Loading-Unloading Time on Best Solution	64
4.1.3 Worst Case Scenario.....	66
4.1.4 Practical Scenario	69
4.1.5 Waiting Time for Truck in DC-to-DC and Local Distribution	71
4.2 Conclusion.....	74
4.3 Future Work	75
REFERENCES	77
APPENDICES	80
Appendix A	81
Appendix B	101

LIST OF TABLES

Table 2.1 Data Sources	15
Table 2.2 County-wise Population in RDC	18
Table 2.3 Number of Packages Delivered by USPS in 2010.....	20
Table 2.4 Vehicle and Person Trips by Purpose in 2009	22
Table 2.5 Per-Capita Daily Person and Vehicle Trips in RDC by Purpose.....	23
Table 2.6 Industry-wise Commodity Flow in 2007	24
Table 4.1 Optimal Number of DCs and Average Delivery Time for Varying <i>LU</i> and <i>pcd</i>	66

LIST OF FIGURES

Figure 1.1 Package Flow in Current Structure.....	2
Figure 1.2 Package Flow in Proposed Structure.....	3
Figure 1.3 Design Procedure.....	8
Figure 2.1 Hierarchical Representation of Geographic Entities [8]	17
Figure 3.1 Pseudo Code for Generating URN	29
Figure 3.2 Plot of Census Blocks on Map of RDC.....	30
Figure 3.3 Final URN with Census Blocks Connected to Network	31
Figure 3.4 Pseudo Code for Generating PLN and Delivery Time Calculation	34
Figure 3.5 Approximate Vehicle Route Length Estimation in Zone of Arbitrary Shape	36
Figure 3.6 Flowchart for Group Formation	40
Figure 3.7 Group Formation for $pcd = 2.29$	41
Figure 3.8 Group Formation for $pcd = 3.48$	42
Figure 3.9 Estimation of Vehicle Route Distance	43
Figure 3.10 Pseudo Code for Estimating Routing Delays for Clusters/Groups	47
Figure 3.11 Pseudo Code for Estimating Routing Delays for Blocks	48
Figure 3.12 Demonstration of Design of Vehicle Route	49
Figure 3.12 Starting Locations and Allocations of 12 DCs.....	55
Figure 3.13 Improved DC Locations and Allocations after 1 st Iteration	57
Figure 3.14 Best Locations and Network of DCs.....	58
Figure 3.15 Pseudo-Code for Heuristic	59
Figure 4.1 Delivery Time vs. Number of DCs for $pcd = 5.28$ and $LU = 0$	61
Figure 4.2 Delivery Delays vs. Number of DCs for $pcd = 5.28$ and $LU = 0$	61
Figure 4.3 Best DC Locations for $pcd = 5.28$ and $LU = 0$ (Best Case)	62
Figure 4.4 Delivery Time vs. Number of DCs for $pcd = 5.28$ and $LU = 10$	64
Figure 4.5 Delivery Delays vs. Number of DCs for $pcd = 5.28$ and $LU = 10$	65
Figure 4.6 LU vs. Optimal Number of DCs and Average Delivery Time	65

Figure 4.7 Delivery Time vs. Number of DCs for $pcd = 2.29$ and $LU = 10$	67
Figure 4.8 Delivery Delays vs. Number of DCs for $pcd = 2.29$ and $LU = 10$	67
Figure 4.9 Best DC Locations for $pcd = 2.29$ and $LU = 10$ (Worst Case).....	68
Figure 4.10 Delivery Time vs. Number of DCs for $pcd = 3.48$ and $LU = 0$	69
Figure 4.11 Best DC Locations for $pcd = 3.48$ and $LU = 0$	70
(Practical Case)	70
Figure 4.12 Delivery Delays vs. Number of DCs for $pcd = 3.48$ and $LU = 0$	71
Figure 4.13 Exponential Relation between Local Delivery Truck Waiting Time and pcd	72
Figure 4.14 Number of DCs and Average Delivery Time vs. pcd for $LU = 0$	73

CHAPTER 1: Introduction

1.1 Introduction

A public logistics network (PLN) [1] has been proposed as an economic and efficient way of expediting delivery of packages. It is a network of public distribution centers (DCs) which can be utilized by multiple organizations. These organizations can share associated logistics costs. Consolidation of a variety of products and truckload transportation help for mitigation of traffic congestion, saving of fuel, and reduction in labor costs. Use of advanced information systems in the PLN can improve coordination in different stages of supply chain minimizing uncertainties.

A network of public distribution centers would be constructed as mid-stage between retailers and customers. Locations of DCs can be in proximity of highways and closer to populated areas in order to expedite package deliveries. The type of products that can be handled by these DCs include furniture, consumer electronics, paper products and stationary items, drugs, textile products, perishable and non-perishable food products, non-food grocery products, home improvement and hardware items. Since these products include almost everything related to domestic and office needs, implementation of the PLN can reduce daily vehicle trips for purpose of shopping by substantial amount. Moreover, not only retailers, wholesalers and manufacturers can use of PLN but third party logistic companies working with cooperative contract, small and midsized freight transportation enterprises can

collaborate with PLN and reduce the costs with high quality service to their customers. Figure 1.1 shows the material (package) flow in the current structure in which customers visit retailers to buy the required products.

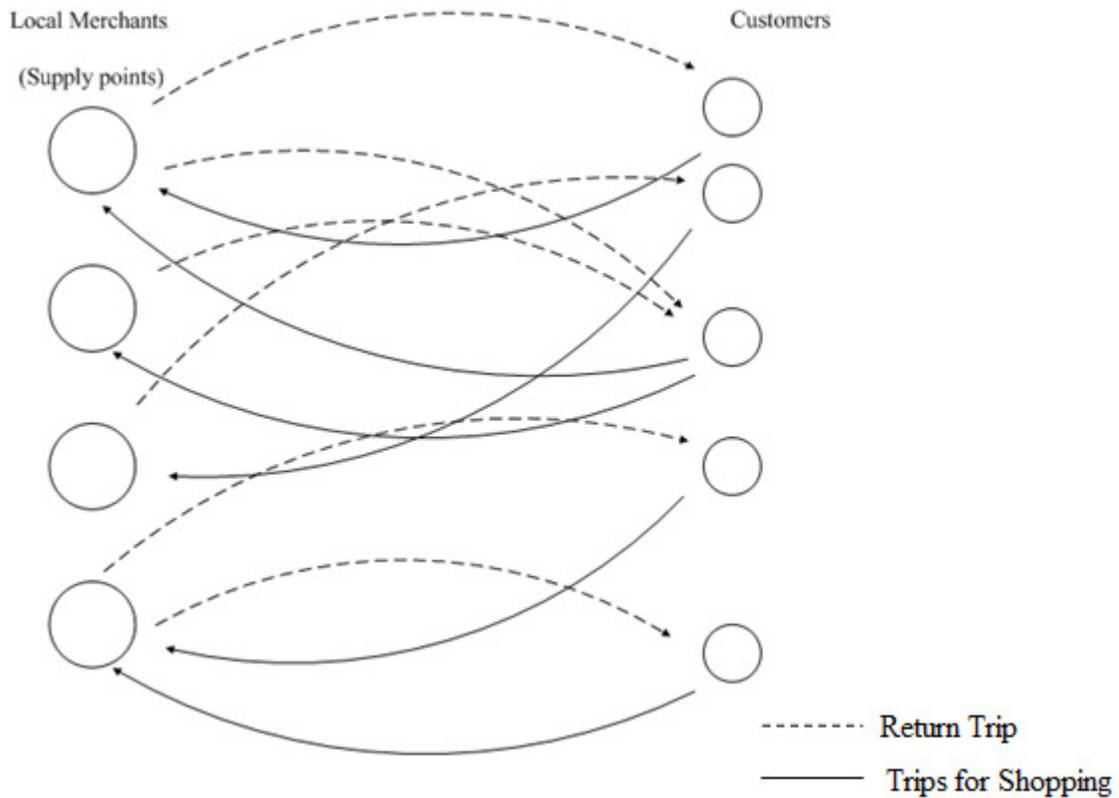


Figure 1.1 Package Flow in Current Structure

The working principle of a public logistics network is very similar to the Internet. The Internet is a network of routers. When a user requests for a data packet, the request is transferred to the respective server. The data packet sent by a server travels through series of routers in the network before reaching to the user. Furthermore, the Internet is not owned by any organization, while organizations connect their own local network to the Internet.

Routers and data packets in the internet are analogous to DCs and packages in a PLN, respectively. Additionally, the entire PLN is not owned by any single organization. There are some organizations having their own established logistics network in the market which can make their business operations very cost-effective. However, not all organizations may be able to afford to build their own network because the capital cost for constructing logistics network infrastructure is too high. Utilizing the features of a PLN could help all the companies in the network reduce their costs and improve freight-market competitiveness. Figure 1.2 shows package flow in the proposed structure.

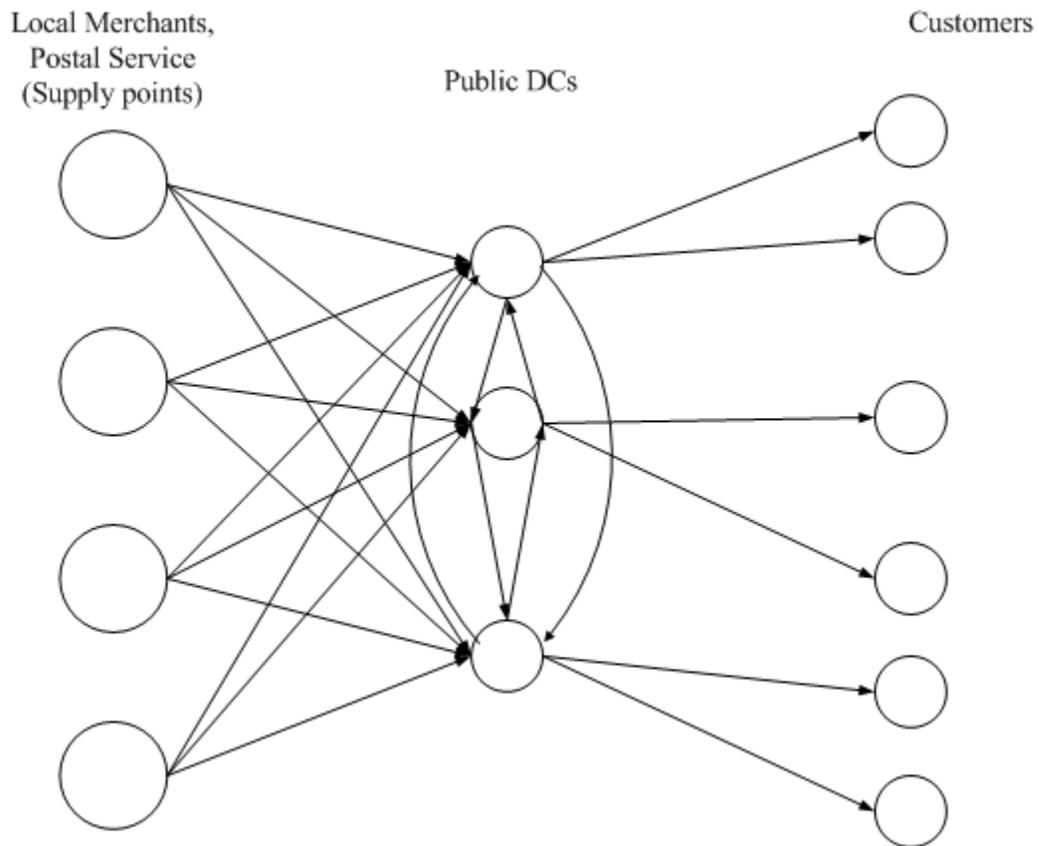


Figure 1.2 Package Flow in Proposed Structure

1.2 Objectives

A procedure for designing a PLN covering the continental U.S. was proposed by Bansal [2]. In this PLN, the average transport time of package was close to nine hours. This research extends the idea of a PLN to handle day-to-day domestic and commercial needs. This purpose can be served by implementing the PLN in metropolitan areas where daily package needs can be fulfilled in two to three hours, although the practical feasibility of a PLN implementation highly depends on daily demand arising in a given region. Considering these points, a task of designing a PLN for the Raleigh-Durham-Cary region of North Carolina (RDC) is undertaken in this thesis.

The main objectives of the thesis are:

1. Compiling the information about per capita daily package demand in RDC. Data collection also includes information regarding the number of vehicle and person trips required for the purpose of shopping at present in RDC.
2. Designing a PLN in RDC by determining the number of DCs and their locations that minimize average package delivery time for different sets of parameter values like per capita demand and loading-unloading time. Also, determining whether design of a PLN is feasible for the lowest value of RDC per capita daily demand and, if not feasible, then estimating the minimum value required to implement a cost effective PLN.

This research is an extension to the Bansal's work [2] of designing a PLN in the continental U.S. in terms of designing vehicle routes and schedules for the purpose of effective distribution of packages directly to customer locations. Vehicle routing helps in estimating local transportation delays, which subsequently can be added to other delays to estimate average package delivery time more accurately.

1.3 Background and Overview

In this section, previous work on PLN design and the tasks undertaken in this thesis to achieve mentioned objectives are discussed. As mentioned in section 1.2, Bansal [2] developed a design approach for PLN. The same design approach can be adopted with some changes in order to generate a PLN in RDC.

Bansal [2] plotted 3-digit ZIP codes on map of the continental U.S. and generated the underlying road network (URN) of interstates and U.S. highways covering the continental U.S. Each ZIP code is associated with some population and hence population points (ZIP codes) were connected to URN. Bansal used functions developed by Kay from the Matlog toolbox [11] to create the URN. Each intersection point in the URN was considered as a potential location for a DC in the PLN. Bansal located a certain number of DCs at key intersection points and created a network of DCs using the Delaunay Triangulation technique [11]. Thus there are two networks that are required to be considered: the URN and the network of DCs (or reduced network). Shortest time paths and distances between each pair of points in the URN and the network of DCs (reduced network) were calculated by using

Dijkstra's algorithm [11]. Average transport time of a package in network of DCs was calculated as the sum of average DC-to-DC travel time, waiting time for a truck, and loading-unloading time. Local travel time from a DC to a customer location was ignored. Afterwards, Bansal used the genetic algorithm to find locations of DCs that minimized the average transport time of a package in the network of DCs. In this analysis, total package demand arising in a day was considered equal to the daily volume of packages delivered by the United Parcel Services (UPS) in the continental U.S.

Differences in Bansal's design approach and the approach used in this research arise for variety of reasons. RDC is very small area as compared to continental U.S. and the proposed PLN for RDC would be handling retail and other types of daily demands. In Bansal's design, local travel time was not too large to compare with other delays. But in the RDC region, local travel time from DC to customer should be considered. Local travel time can be estimated more accurately if fine resolution of population, i.e., the exact locations of customers in RDC, is known. Since very few 3-digit ZIP codes cover RDC, they are not suitable as demand/population points since they will not give accurate results for local travel delays due to lack of fine resolution of population. Hence the lowest geographic entity, the census block, was used to achieve finer distribution of population.

Local travel delay (local package delivery delay) cannot be determined by the direct addition of weighted travel time between a DC and a customer location because packages can be delivered in an effective way by visiting multiple customers in single trip. Hence, multi-stop vehicle routes were required in order to estimate local travel delays or local delivery times in

the more practical and accurate way. Furthermore, UPS package delivery data is not a correct input to the analysis because daily domestic and commercial needs (like groceries etc.) are not delivered by UPS. Daily per capita package demand in RDC will also be estimated in this thesis. Figure 1.1 shows the overview of the complete design procedure.

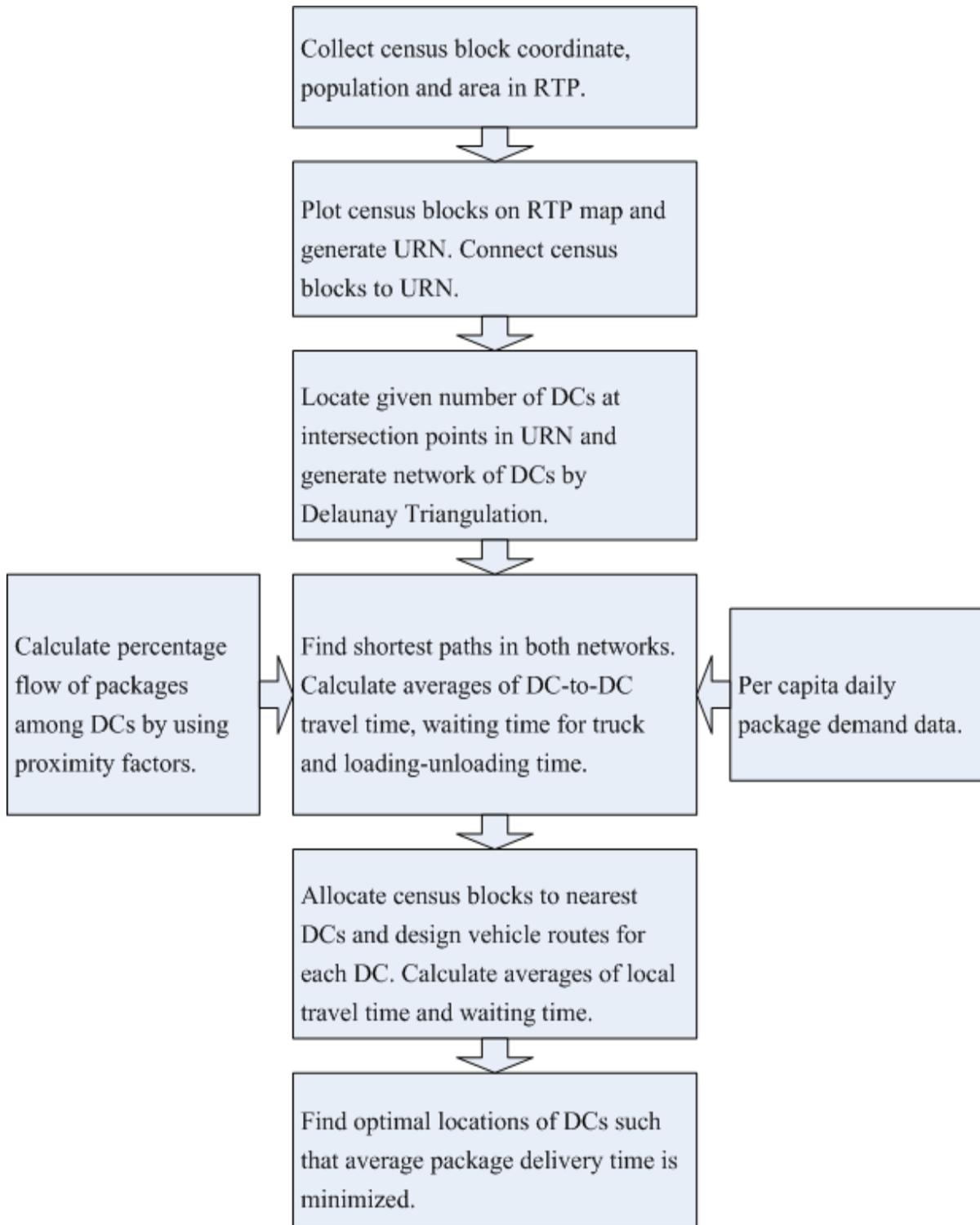


Figure 1.3 Design Procedure

1.4 Literature Review

This research specifically requires knowledge of the working principles of a PLN, developments in network design approaches, queuing systems, discrete location models, and the vehicle routing problem. Bansal [2] would be the most useful reference for PLN design approach but since section 1.3 covered all details of PLN design this section mainly concentrates on other relevant literature excluding Bansal.

1.4.1 PLN Design

Multi-organization usage of public warehouses was further extended into the concept of a public logistics network in Kay and Parlikad [1]. By using advance information technology systems, information can be updated at all locations and stages in PLN to enable pricing negotiations between customers. Kay and Parlikad developed the 36-DC network in the southeastern U.S. Average transport time of a package in the PLN was compared with the average package transport time in a hub-and-spoke and a point-to-point network. It was observed that packages can be transported in minimum time using the PLN when the loading-unloading time for packages is relatively short.

In terms of analysis, the significant development by Kay and Parlikad [1] is the derivation of order based proximity factors. Proximity factors are an indication of the degree to which a DC is more likely to transport packages to nearby DCs as opposed to DCs located further away. Proximity factors are used to estimate the flow among DCs based on population and by ranking DCs in ascending order of distances instead of using actual distance values.

In Bansal [2], the waiting time for the truck was calculated as half of the truck headway. In practical situations, the headway ratio is greater than 0.5. The queuing of packages at DCs was studied in detail, and average waiting time of packages in two different queues was estimated analytically in Ling [3]. Ling estimated a more precise value for the headway ratio of 5.5.

1.4.2 Design of Vehicle Schedule and Routes

Design of a PLN for RDC requires estimation of the local transportation time along with other delays because, for a PLN in the region as small as RDC, local transport time is comparable to other delays. Since a delivery vehicle operates with full package carrying capacity, it carries packages of various customers and has to make multiple stops on its delivery route.

The vehicle routing problem (VRP) and travelling salesman problem (TSP) are similar in nature and both are NP-hard problems. Different algorithms and analytical methods have been developed to solve these problems. Some of the algorithms' performance depends on the number of cities to be visited. Solutions are simple and quick with a small number of cities visited. The complexity of the problem increases with: number of cities, the number of vehicles is unknown; routes are subject to capacity constraints, and routes are subject to time windows. In many cases, the solution approach depends on the metric type, for example, Euclidean or Manhattan.

Exact algorithms include branch and bound algorithm, cutting plane method or some techniques reminiscent of linear programming. But again performance of these methods depends on the complexity of the problem. Exact algorithms take a large amount of time for computation, so approximate algorithms or heuristics have been developed. The nearest neighbor algorithm is a greedy procedure. If the nearest unvisited city is visited one after another to form a route, then this route distance would be approximately 1.25 times greater than optimal distance [12]. However, for some configuration of cities, nearest neighbor algorithm gives the worst solution. Christofides' algorithm forms a minimum spanning tree of the cities. This tree is converted into an Eulerian circuit and then transformed into a Hamiltonian circuit using a shortcut technique. This is the effective heuristic with approximation ratio of 1.5. It is important to note that these heuristics can give expected minimum route length but not the actual route. The closest unvisited city algorithm finds numerous paths by taking different starting points and visiting the closest unvisited city. On each of these routes one-way switches, two-way switches, three-way switches and so on are performed to find the path with minimum length. The complexity of this process is again function of number of cities in the map [20].

As opposed to the above procedures, continuum approximation methods are simple and effective. Using the distribution of cities on a map and the geometry of the map, the approximate length of optimal route can be estimated. These methods can also help in designing the routes graphically. In Larson and Odoni [4], the analytical expression for approximate tour distance in a square-shaped region was derived. The distance has two parts:

line haul distance from depot to random city in region and detour distance, which is average distance between any two cities visited consecutively.

The equation for tour distance per point [6] is given below:

$$\frac{L}{N} = 1.8\rho \left[\frac{1}{C} + \frac{1}{\sqrt{N}} \right],$$

where,

L = tour distance,

N = number of cities,

C = number of stops per vehicle,

ρ = line-haul distance from centrally located depot to any random city in the square.

This equation is applicable to Euclidean metric regions with uniformly distributed cities. This research was extended in Daganzo [5], [6], [7] to regions of arbitrary shapes with other types of metrics. The equation derived by Daganzo contains the tour distance in same two parts as in above equation with different values of coefficients. The coefficients in Daganzo's equation were derived from the assumption that $C > 6$ and $N > 4C^2$.

Chien performed simulations and linear regression to modify coefficient values in Daganzo's equation for rectangular sub-region with a slenderness factor varying from 1 to 8. Daganzo [5],[6],[7] proposed an intuitive analytical model for VRP with time window constraints with identical customers and customers with varying demand with number of vehicle schedules known *a priori*. This model requires knowledge of demand patterns.

In this thesis, Daganzo's continuum approximation approach is used because it is simple, near-optimal, quick, and expected tour distance is calculated without actually graphically designing the routes. Details explanation of this approach is given in chapter 3.

1.5 Thesis Outline

Chapter 2 of thesis discusses data collection related to daily package demand in RDC, vehicle and person trips, average package weight and consumption of various commodities in the RDC. In Chapter 3, the solution approach for design of the PLN and the heuristic are developed. Chapter 4 discusses the results of analysis and conclusions of the research along with the possible scope for future work.

CHAPTER 2: Data Collection

2.1 Introduction

The package-demand arising from the population in RDC drives the material flow in the PLN. The number of DCs required to locate in the PLN depend on the daily demand of packages. So, it is essential to find the number of packages required in RDC per day in order to check the feasibility of the PLN implementation in RDC. This chapter discusses the different sources used to compile demand data, followed by a description of the data collected. This is divided into two parts: geographic entity-wise distribution of population in RDC and estimation of the per-capita daily package demand.

2.2 Data Sources

The data required for the design procedure was collected from multiple sources. The main source for extracting demographic data in the U.S would be the U.S. Census Bureau. The package demand data is estimated by collecting facts and figures from several sources although the most relevant information about package demand and commodity flow can be found from the Bureau of Transportation Statistics and the U.S. Department of Transportation. The sources with their respective information topics are shown in table 2.1.

Table 2.1 Data Sources

Sr.	Source	Information
1	2010 census geographic terms and concepts (pdf file)	Hierarchical representation of geographic entities, definition and method of formation of entities, numeric representation, counts by state
2	2010 US census Gazetteer files	Geographic entity-wise (except for census block) information about GEOID, land area, water area, longitude and latitude
3	2010 US census centers of population by census block group	State-wise population, longitude, latitude and numeric codes for census blocks
4	2010 US census block relationship files	Census 2000 Tabulation Block to 2010 Census Tabulation Block files containing information about land and water areas of census blocks.
5	Statistical abstract of the United States: 2011	Shipment characteristics by mode of transportation (average shipment miles, ton-miles and tonnage), Urban area wise daily vehicle miles and total annual delay due to congestion, vehicle miles by type of vehicle
6	Data and statistics by Research and Innovative Technology Administration, Bureau of Transportation Statistics	2007 Commodity Flow Survey by mode of transportation, commodity and NAICS
7	2009 National Household Travel Survey (frequently asked for tables) http://nhts.ornl.gov/tables09/FatCat.aspx	Data about person trips, vehicle trips, person miles, vehicle miles by purpose of travel and mode of transport
8	Food and Marketing Institute (FMI)	Supermarket facts
9	USPS.gov	USPS delivery truck specifications, 2010 annual report about mail delivery
10	UPS press room fact sheets	UPS delivery truck specifications

2.3 Geographic Boundary Limits of RDC

Raleigh-Durham-Cary, as defined for statistical purposes as the Raleigh-Durham-Cary combined statistical area (CSA). This area comprises eight counties, although in 2003 the U.S. Census Bureau divided the region into two metropolitan statistical areas and one metropolitan area. The counties constituting this area are Chatham, Durham, Franklin, Granville, Harnett, Johnston, Orange and Wake.

Selecting a geographic entity and locating population centers are followed by generation of an Underlying Road Network (URN) in RDC. The geographic population centers will then be connected to this URN. These population centers will generate package demand which drives material flow in the network. In RDC, it would be difficult to find a finely resolved distribution of the population, although the U.S. Census provides the population for a given level of geography and the exact center of mass of the population. Thus, the appropriate level of geography (geographic entity) should be chosen which can be considered for further analysis. The highest possible level of geography would be a '*county*', which would give a total of eight points representing population centers of the counties mentioned above. The population in RDC can be finely resolved by considering some lower geography levels.

The data can be collected by dividing it in two categories:

1. Collection of demographic data for different levels of geography.
2. Per-capita daily package demand in RDC.

2.4 Collection of Demographic Data for Different Levels of Geography

U.S. Census has released multiple 2010 Census Data Products [13]. Each of these data products provides the data with a certain level geographic detail. The levels of geographic detail follow a hierarchical structure that is derived from the legal, administrative or area relationships of the geographical entities. Figure 2.1 represents the hierarchical relationship of census geographic entities.

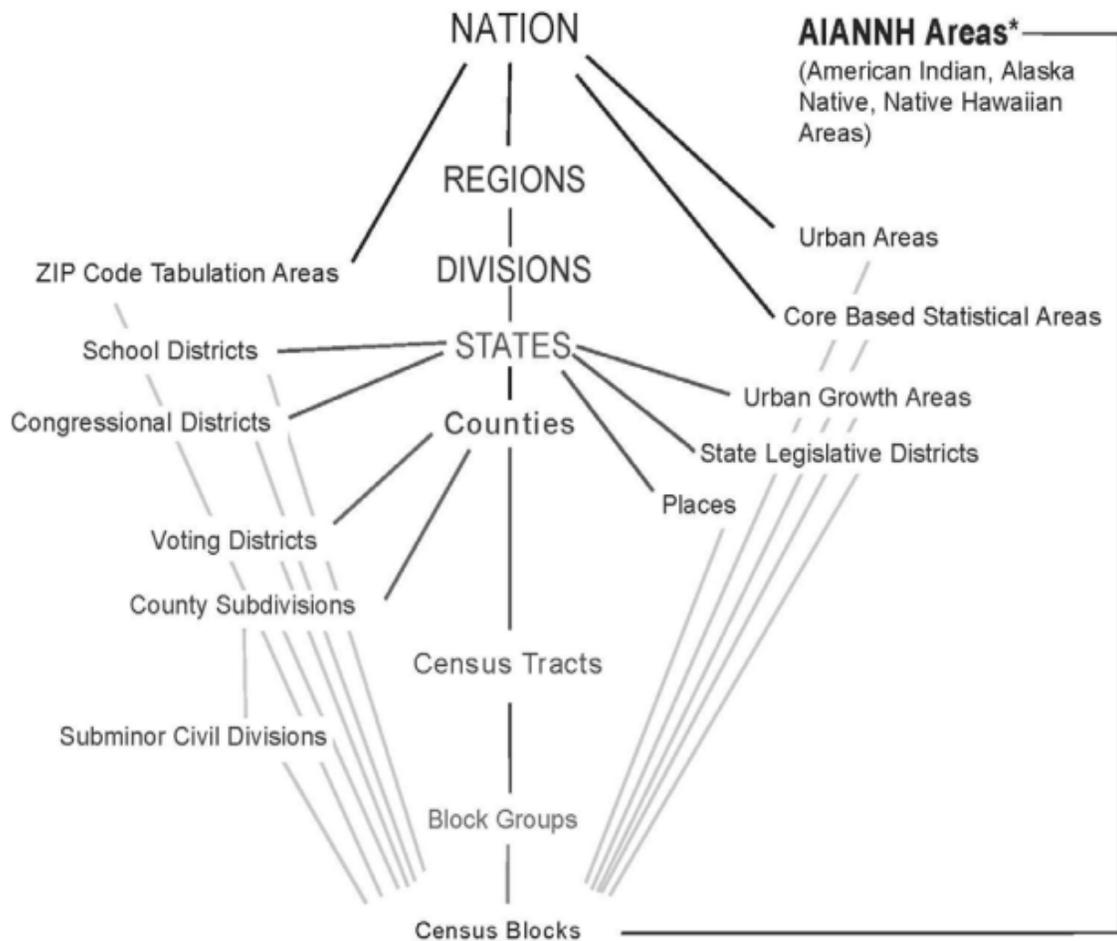


Figure 2.1 Hierarchical Representation of Geographic Entities [8]

The demographic data on levels of five-digit ZIP codes, census tracts and census blocks was collected. RDC contains a total of 96 five-digit ZIP codes, 365 census tracts and 925 census blocks. The data products provided by the U.S. Census contain various features of geographic entities such as population, land area, longitude and latitude of population centers with numerical codes assigned to each geographic entity. For instance, the numerical code 37201031 indicates block 1 in the census tract 20103 in Chatham County designated by county FIP 37. Total population in RDC in 2010 was 1,769,977. Since total population was almost 1.8 million, it was decided to use the census blocks as a geographic entity for analysis in order to achieve finer resolution. Hence total 925 blocks will be plotted on the map of RDC as demand points to design the PLN.

Table 2.2 County-wise Population in RDC

County	Population	County	Population
Chatham	63,505	Harnett	114,678
Durham	267,587	Johnston	168,878
Franklin	60,619	Orange	133,801
Granville	59,916	Wake	900,993
TOTAL	1,769,977		

2.5 Per Capita Daily Package Demand in RDC

The PLN will be designed to enable the delivery of a package to customer locations in the minimum possible time, subject to efficient delivery vehicle operation. The PLN will consider both types of package demands. USPS, UPS, FedEx and other courier companies deliver packages to the customer locations and customers fulfill their other requirements by traveling to grocers, retailers or other merchants. Since this number is imprecise, some lower and upper bounds on the package demands will be estimated such that package demand will vary within the estimated lower and upper bounds.

The estimated total number of packages delivered to the customer locations by the courier companies is added to the package demand that arises through customer shopping trips for business services, consumer, and grocery products. In an area as small as RDC, USPS service is primarily used to mail packages because UPS or FedEx is a more expensive option as compared to USPS. Hence, instead of considering the total package deliveries by all the courier services, only USPS package deliveries are taken into consideration.

2.5.1 Package Deliveries by USPS

USPS provides total quarterly and yearly sales data on their official website. This data is expressed in terms of revenue, number of pieces and weight in pounds. In this case, the total number of pieces delivered by the USPS in year 2010 can be considered. In 2010, USPS had delivered almost 170 billion pieces in the entire U.S. There are multiple services provided by USPS and very few of them deliver packages/parcels. Therefore, it is essential to filter this

data by subtracting the number of pieces those were postcards, letters, periodicals, magazines, international mails etc. Thus number of packages delivered by USPS in year 2010 was estimated to be 3,334,682,000. Category-wise number of pieces is shown in the table 2.3.

Table 2.3 Number of Packages Delivered by USPS in 2010

USPS segment	Pieces (in thousands)
Parcels by standard mail	682,403
Parcels by first class mail	574,429
Shipping Services	1,419,584
Package Services	658,266
Total	3,334,682

In 2010, the U.S. population was 309,050,816, while the RDC population was 1,769,977.

Hence fraction of the total U.S. population in RDC in 2010 was 0.0057.

Package demand in RDC fulfilled by USPS:

Delivery of packages in RDC = $0.0057 \times 3,334,682,000 = 19,098,187.54$ in 2010.

Daily per-capita package demand in RDC = $\frac{19,098,187.54}{1,769,977 \times 365} = 0.03$ in 2010

One person gets delivery of one package by USPS in almost 30 days or one month.

2.5.2 Package Demand Fulfilled by Customer Trips

The part of package demand besides the above USPS value of 0.03 per-capita per-day constitutes the number of packages purchased by the customers through visits to local merchants. This part of the data requires information regarding the number of vehicle and person trips by customers for multiple purposes and by various modes of transportation. The National Housing Travel Survey 2009 (NHTS) [14] provides this type of information. The travel data was collected from the sample of the U.S. population representing the entire U.S. and hence contains people from different groups based on cultural diversity, gender, and age group. This data was statistically analyzed to generalize the information for the entire U.S. Although this survey was conducted in 2009, the same data can be used to estimate the package demand because, it is assumed that the value of per-capita daily demand will not likely change significantly from 2009 to 2010. Moreover, the customer trip data is available in units of vehicle trips as well as person trips. Although per-capita package demand can be estimated more accurately using person trips data, vehicle trips data is also shown in following table. Table 2.4 shows the total surface trips for the purpose of shopping and meals in 2009.

Table 2.4 Vehicle and Person Trips by Purpose in 2009

	Shopping	Meal	Total
Person Trips in millions	68713	26483	95196
Vehicle Trips in millions	45201	14595	59792

Daily shopping person-trips (in millions) can be calculated as,

$$\text{Daily person trips for shopping} = \frac{68713}{365} = 188.25$$

Daily person trips for shopping in RDC = $188.25 \times \text{Fraction of total U.S. population in RDC}$

$$= 188.25 \times 0.0057 = 1.073058$$

If daily person trips of 1.07 million is divided by the RDC population in year of 2009, i.e., 1,749,947; it will give a per-capita daily person trips in RDC for shopping equal to 0.61.

In the same way, values in the Table 2.5 were calculated. Considering each person trip for purpose of shopping equivalent to one package would be an incorrect assumption because in a shopping trip, a person might visit more than one shop and hence number of packages per trip is still an unknown parameter. Also, it is clear that the meals cannot be fulfilled through the PLN, so trips only for the purpose of shopping will be considered.

Table 2.5 Per-Capita Daily Person and Vehicle Trips in RDC by Purpose

	Shopping	Meal	Total
Per capita daily Person Trips	0.61	0.24	0.85
Per capita daily Vehicle Trips	0.41	0.13	0.54

According to the Food Marketing Institute's (FMI's) 2009 estimates [15], a person visits a supermarket 2.1 times in a week. Hence the daily supermarket trips in RDC are 528,836 and per-capita daily supermarket trips in RDC are 0.3. This means almost 50% of the shopping trips per person are for the purpose of grocery shopping at the supermarkets. If a customer visits more than one supermarket in one trip then according to FMI's supermarket data, trips are equal to the number of supermarkets visited by the customer. This observation is important because, each supermarket trip can be considered equivalent to one package. Therefore, per-capita daily package demand for grocery from the supermarkets will be 0.3. Besides grocery, there are other product requirements that should be added to the 0.3.

Now it is essential to find the relationship between person trips and the number of packages per trip. For this estimation, the Commodity Flow Survey 2007 data [16] is used. The industry-wise material flow in the U.S. and RDC is tabulated below. Total tonnage flow in RDC is calculated by multiplying the tonnage flow in the U.S. with the fraction of the total U.S. population residing in RDC (0.0057).

Table 2.6 Industry-wise Commodity Flow in 2007

NAICS	Industry	Thousands of Tons in USA	Thousands of Tons in RDC
4232	Furniture & Mattresses	20,845	114.02
4236	Electrical & Electronics	18532	101.37
4237	Hardware Wholesale	23979	131.17
4239	Miscellaneous Durables	359,428	1,966.07
4241	Paper Products	41,218	225.46
4242	Drugs	21,188	115.90
4243	Apparel	7,279	39.82
4244	Grocery	314,826	1,722.10
4541	E-shopping	2,740	14.99
	TOTAL	810,035	4,430.89

Recall that per-capita daily package demand from the supermarkets is 0.3 and hence the total daily package demand in RDC will be 528,836. In the supermarkets, other types of products are also sold apart from grocery, but FMI estimated its data considering only the perishable products/food products and the non-food grocery products. Therefore, it takes 528,836 daily

trips/packages for consumption of 1,722,100 tons of grocery per year in RDC. Thus the average weight of grocery package would be 3.26 kg.

Now, if per-day per-capita consumption (in kilograms) is divided by 3.26 kg, it will give approximate daily package demand in RDC. The annual consumption of commodities in RDC is 4,430,890 tons. Hence daily per-capita consumption in RDC is 7.36 kg. Thus,

Daily per capita package demand = Daily per capita consumption/Average weight of package

$$= \frac{7.36}{3.26} = 2.26$$

Also, the average package weight can be estimated using the total pieces delivered by the USPS in 2010 and the total weight of those pieces.

Average package weight using USPS data = weight of pieces / no. of pieces

$$= \frac{9,670,578}{3,334,682} = 2.9 \text{ lb} = 1.4 \text{ kg.}$$

Using this value,

Daily per-capita package demand= Daily per capita consumption/Average weight of package

$$= \frac{7.36}{1.4} = 5.25$$

In the further analysis, demand arising from the shopping trips is assumed to vary in the interval [2.26, 5.25]. After adding number of packages delivered by USPS per-day per-capita

of 0.03, the final range for the per-capita daily package demand in RDC would be [2.29, 5.28]. Since this interval represents two extreme ends of per-capita demand, the most likely value of per-capita demand can be considered to be geometric mean of the upper and lower bound, i.e. 3.48.

2.6 Package Carrying Capacity of a Vehicle

The number of packages delivered by a truck in one trip will depend on vehicle capacity and average size of a package. UPS uses a composite car CV-23 and P70 to deliver packages to customer locations [17]. The cargo sizes of these vehicles are 630 cubic feet and 700 cubic feet respectively. The average weight of a package and the maximum payload of the vehicle are not considered here because in most of the cases, the shipment surpasses a vehicle volume capacity before its weight capacity. According to UPS and USPS data, a delivery truck can carry around 300 packages [17]. With load factor of 0.8 and 0.6, the number of packages a vehicle can carry in a single trip will be equal to 240 and 180, respectively. Although the physical package-carrying capacity of a delivery truck is 300 ft³, it cannot be fully utilized due to several reasons for instance, irregular sizes of packages, waiting of a truck for additional time until the cargo is completely filled etc. Therefore, effective capacity of a truck is considered to be varying from 60% to 80% of truck's physical package-carrying capacity.

CHAPTER 3: Design of the PLN

3.1 Introduction

In Bansal [2], a PLN for the continental U.S. was designed. This research uses the same design approach that was developed by Bansal. Bansal's design can be explained as four step process that includes generation of the Underlying Road Network (URN), developing the network of public DCs, estimation of average package delivery time, and finding public DC locations that minimize average package delivery time. In the RDC, the PLN will be designed using a similar four step process, with some modifications in the last two steps. This chapter briefly discusses the first two steps, followed by a detailed discussion of the modifications of the last two steps.

3.2 Generation of the URN and the Network of Public DCs

The population in RDC is represented by total 925 U.S. census blocks that are plotted on the map of RDC as shown in figure 3.2. A sub-graph of the road network was generated that is then followed by the removal of two-degree nodes from the network. The arcs connecting census blocks to the URN are added to the network and the shortest time paths and distances between each pair of points are calculated using Dijkstra's algorithm [18] [11]. Pseudo code for the URN generation is given in Figure 3.1. Figure 3.3 shows the URN with census blocks connected to it.

Each point in this network is a potential location for a DC. DCs will be located at some of the key points and then connected to each other using Delaunay Triangulation [19] to form a network of public DCs. The shortest time paths between all pairs of DCs is found and those paths and distances are then used to calculate the percent flow of the packages, w_{ij} from DC i to DC j using order based proximity factors developed by Kay and Parlikad [1] using following equations.

$$w'_{i[j]} = w_{i[j]}^0 \cdot \frac{\frac{p}{m} \left(1 - \frac{p}{m}\right)^{(j-1)}}{\frac{1}{m} \sum_{k=1}^m \frac{p}{m} \left(1 - \frac{p}{m}\right)^{(k-1)}} = w_{i[j]}^0 \cdot \frac{\left(1 - \frac{p}{m}\right)^{(j-1)}}{\frac{1}{m} \sum_{k=1}^m \left(1 - \frac{p}{m}\right)^{(k-1)}}$$

$$w_{i[j]} = \frac{w'_{i[j]}}{\sum_{k=1}^m \sum_{l=1}^m w'_{kl}}$$

If w_i is DC_i 's percentage of total population then $w_{ij}^0 = w_i \cdot w_j$. For calculating w_{ij} for DC_i , total m DCs in the network, $DC_{[1]}$, $DC_{[2]}$,, $DC_{[m]}$ are ordered in terms of increasing great circle distance from DC_i . The average distance travelled by a LTL shipment in the continental US is 750 miles. Kay estimated value of the proximity factor $p = 2.55$ such that average distance travelled in the PLN, developed in the continental US, is 750 miles. In the RDC, average distance travelled by a package should not be more than 4-7 miles. Hence p is estimated equal to 19.55 for which average distance travelled by a package lies in the range of 4-7 miles.

There will be flow of material between DCs in the PLN and DCs located outside the RDC. Considering this material flow, four DCs were located at 3-digit ZIP codes on the periphery

of the RDC in four different directions. A total of 15% of the package demand is assumed to be flowing outside the RDC.

```
1. Load (coordinates XY, population and areas of census blocks).
2. Subgraph(XY) % Generate URN of interstates and roads covering census blocks and
    % save nodes XY and arc origin i, destination j and distance d as IJD.
3. (distance in IJD) / speed % Calculate travel times IJT.
4. Thin (URN) % Eliminate 2nd degree nodes and accordingly modify XY, IJD and IJT.
5. Addconnector (blocks, URN) % Add connectors from blocks to nearest nodes on
    % network.
6. (distance in IJD) / speed % Calculate travel times IJT in new network.
7. Dijk (URN) % Find shortest paths between each pair of nodes and save paths P and times
    % T using Dijkstra's Algorithm.
8. Calculate distance D of shortest time routes P.
9. Save nodes XY, IJD and IJT matrices of final network.
10. Save shortest paths P, distances D and times T.
```

Figure 3.1 Pseudo Code for Generating URN¹

¹ Highlighted words are keywords or functions from Matlog/Matlab/Appendix A

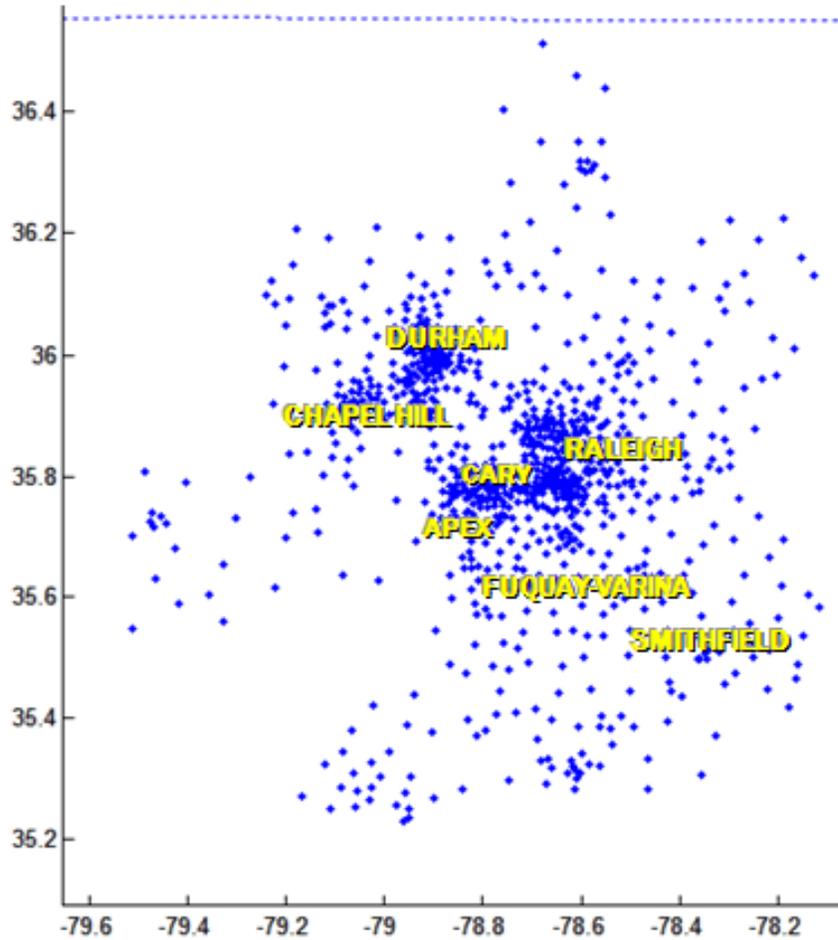


Figure 3.2 Plot of Census Blocks on Map of RDC

3.3 Calculation for Average Package Delivery Time

When a customer places an order, package availability is checked at the closest DC. If a package is not available there, then it is transferred to the closest DC from another DC in the network via its shortest path. Thus, before a package arrives to the closest DC, it might travel through multiple DCs. The package will be loaded and unloaded at each DC and it has to wait in queue for an available truck, which can then pick it up for further transport. Once a

package reaches to the closest DC, it again has to wait for an available delivery truck that will deliver it to the customer location.

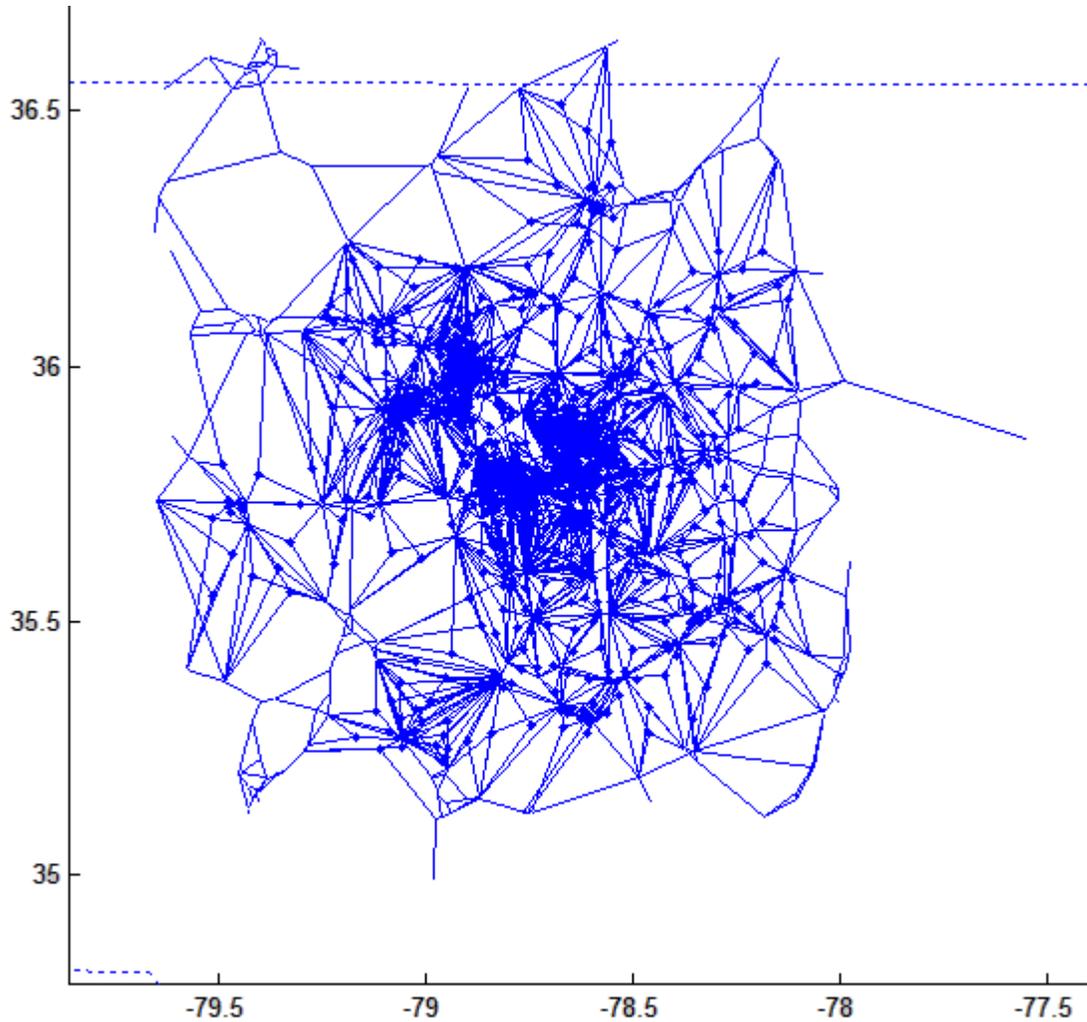


Figure 3.3 Final URN with Census Blocks Connected to Network

Hence, a package delivery time constitutes four parts: waiting time for truck, DC-to-DC transportation time, loading-unloading time, and local transportation time from a DC to a customer location. In the PLN design for the continental U.S., Bansal [2] considered all parts

except the local transportation time. The flow of package demand along each arc in the network of DCs can be calculated using w_{ij} . These demands, when divided by a truck's package-carrying capacity, will give the average number of trips on each arc. Thus if service time is 14 hours per-day then truck headway on each arc would be the ratio of 14 and the number of trips. Waiting time for an available truck is the product of truck headway and headway ratio 1.7. Bansal [2] used lower bound value of 0.5 for headway ratio while Ling [3] estimated headway ratio of 5.5 in the worst case scenario. This research uses the most likely value of 1.7 which is the geometric mean of 0.5 and 5.5, for headway ratio. DC-to-DC transportation time is the sum of the time required to travel between each pair of DCs along the shortest path. Furthermore, this analysis calculates the package delivery time for three different values of loading-unloading times: 0, 5 and 10 minutes.

In the PLN, customers will be allocated to the nearest DC. Hence, population in RDC can be divided into clusters of census blocks with only one DC allocated to all the blocks in that cluster. In a cluster, a delivery truck has to visit each block multiple times because demand of the packages coming from the blocks is distributed over time in a day. In a given time window, each block will generate non-zero demand. All the blocks can be visited by the trucks for faster delivery of the packages in that time period, but this will increase the travel distance and time. On the other hand, to reduce travel distance, some blocks can be kept unvisited until demand coming from those blocks reaches a critical level by allowing some increase in the waiting time of packages for available truck. This decision cannot be made arbitrarily; thus, the vehicle routes and schedules were designed in order to increase

efficiency of the package distribution operation from the DC to the customer locations inside its cluster. Section 3.4 describes the approximate vehicle route length estimation procedure in detail. Pseudo code for generation of the network of DCs and average delivery time calculation is given in Figure 3.4.

```

1. Load (nodes XY, IJD, IJT, shortest paths P, times T and distances D)
2. Allocation (blocks) % Allocate blocks to the closest DC
3. Delaunay (DCs) % Delaunay triangulation to create network of DCs.
4. Calculate IJDT matrix containing arcs, distances and times in network of DCs.
5. Dijk(network of DCs) % Calculate shortest times sT and paths sP in network of DCs.
6. Find distances sD of shortest paths sP in network of DCs.
7. For all pairs of uncommon nodes in adjacent triangles in network of DCs
If distance between nodes in URN < 0.85*distance between nodes in network of DCs
    Add direct arc between those two nodes.
End
End
8. For all triangles in network of DCs
    If [largest side > 0.9*(sum of other two sides)] Remove largest side End
End
9. Dijk(network of DCs) % Find shortest times sT and paths sP in modified PLN.
10. Find distances sD of shortest paths sP in modified network of DCs.
11. Proxfac(sD,DCs) % Finds percentage flow of demand among DCs.
12. DemArc(sP) % Calculate demand flow on each arc in reduced DC network.
14. Calculate trips and hence headway on each arc.
15. WFTT(sP, headway) % Waiting time for truck.
16. LoadUnload(LU,sP) % Loading-unloading time of packages.
17. DC2DCTT = sT % DC-to-DC transport time.
18. VRP (DCs, allocations) % Local transport time and waiting time.
19. sum (waiting time, DC-to-DC travel time, loading-unloading time, local travel time)
    % Network cost or average delivery time is calculated as sum of all delays.

```

Figure 3.4 Pseudo Code for Generating PLN and Delivery Time Calculation

3.4 Vehicle Routing from DCs to Customer Locations

When packages arrive to the DCs that are closest to their customer locations, they are delivered to the customers by designing the vehicle routes and schedules. Allocation of the census blocks to the nearest DCs generates clusters in RDC and the number of clusters is equal to the number of DCs located in the PLN. Hence, vehicle routes are required to be designed for each cluster separately. The vehicle routes are designed by using a continuum approximation method developed in Daganzo [5] [6] to calculate the expected distance travelled in the vehicle route.

3.4.1 Estimation of Approximate Vehicle Route Distance

If N customer locations are uniformly distributed in plane of area R , then total N/C tours can be designed with an expected value of travel distance in each tour of

$$\text{Tour Distance} = 2r + \frac{kC}{\sqrt{\frac{N}{R}}}$$

where r = line-haul distance

k = dimensionless proportionality constant depends on metric type (for Euclidean metric, $k=0.57$)

C = number of stops by vehicle per tour = V_{max}/v

V_{max} = effective vehicle capacity, v = average customer demand in a time window.

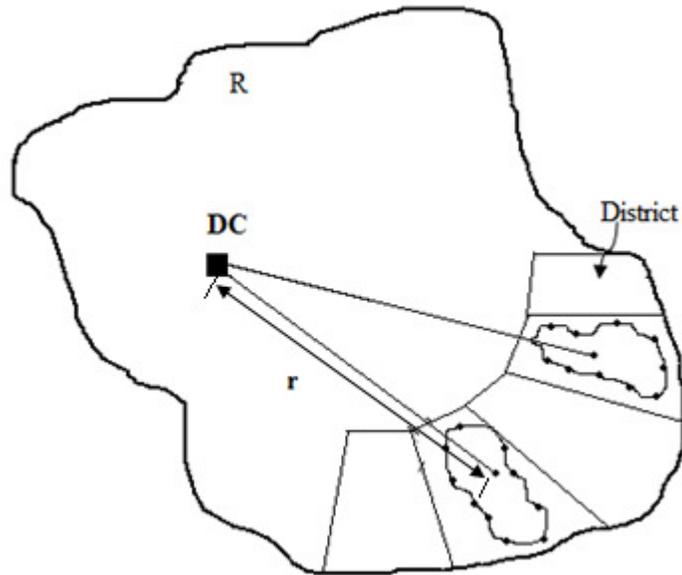


Figure 3.5 Approximate Vehicle Route Length Estimation in Zone of Arbitrary Shape

As shown in the figure 3.5, a cluster of blocks can be divided into districts, and one tour can be designed for each district. First term in this equation estimates the line haul distance required to be traveled, while the second term is C times the average distance between two neighboring points in the district. Although exact customer locations are not available, it can be assumed that the population is uniformly distributed within the area of a census block. Also, it is assumed that all the per-capita per-day number of packages will be ordered and delivered within 14 hours of service time in a 24-hour day.

Vehicle routes cannot be designed considering the total number of packages ordered in a day by all the blocks in the cluster because that will mean packages will only be delivered once

the complete day's orders is received. But the packages should be delivered simultaneously with the placement of orders and hence packages ordered per unit time should be considered as input to the route design procedure. Thus the package demand per hour generated from each block is calculated. If the mean of the per-hour demand values over the number of blocks is v , then the number of blocks (stops) to be visited in each tour C can easily be obtained. The package carrying capacity of a delivery truck is estimated to be varying between 180 and 240. Therefore, it was decided that a separate truck will be assigned for the census blocks that are having per hour demand greater than 180 packages while remaining census blocks are grouped in such a way that each group has per hour package demand greater than 180. In this way, package carrying capacity of the delivery truck can be completely utilized by assigning a separate truck to each group of the census blocks. The assigned truck will visit all the blocks in the group.

For a given value of per capita demand, per hour demand of each census block can be calculated followed by filtering the census blocks that are having per hour demand less than 180. The groups of these census blocks are formed such that each group has per hour package demand greater than or equal to 180. Population centroids, population and area for each group are found. Thus, each group can be considered equivalent to one census block. Hence all the census blocks will have per hour package demand more than 180. The procedure for group formation is called *preprocessing* and it is explained in section 3.4.2. Furthermore; in a given group of blocks, the sum of per hour package demand of all the blocks will be more than 180. This means, formation of the groups depends on the per hour

package demand of the census blocks. Thus formation of the groups is different for different values of per capita demand. The groups are formed for several values of per capita demand and saved.

3.4.2 Preprocessing

For a given value of per capita demand (pcd), per hour package demand generated by each census block is calculated followed by filtering out only those blocks that are having per hour demand less than 180. To form the groups of these filtered blocks, firstly these blocks are connected to each other by Delaunay triangulation technique. The groups can be formed by starting from the periphery of the network and then going into the center. This process can be reversed by starting from the center and then ending to the periphery as well. Also, a census block can be added to a group based on the distance of a census block from the blocks in the group or per hour demand value of neighboring blocks. After trying different alternatives, it was decided that a periphery-to-center approach should be used because grouping is observed to be better in this case.

A census block in the network having the least connectivity is selected and the blocks that are the closest and connected to it are added till total per hour demand of a group goes over 180. A population centroid of the group is calculated. Total population and area of the group is the sum of individual block populations and areas, respectively. The next group is formed by starting with the block that has next least connectivity. The procedure is repeated till there are only those blocks left that are not connected to any other ungrouped block. These blocks are

added to the groups which are closest to them. This completes the preprocessing. It was observed that for $pcd = 3.48$ and 5.28 , there are only 41 and 9 census blocks respectively that are having per hour package demand less than 180. Some of these blocks are too far away from other blocks that they cannot be a part of any group because even if they are added to the closest group, it will increase the vehicle route distance significantly. Hence the vehicle routes can be designed for these blocks by manually grouping them with the blocks with per hour demand greater than or equal to 180. Figure 3.8 demonstrates this kind of incident. Figure 3.6 explains the preprocessing in detail. Figure 3.7 shows groups for $pcd = 2.29$. Total 81 groups from 178 blocks were formed. All 178 blocks are numbered in figure 3.7 such that all the blocks in a given group are assigned the same number.

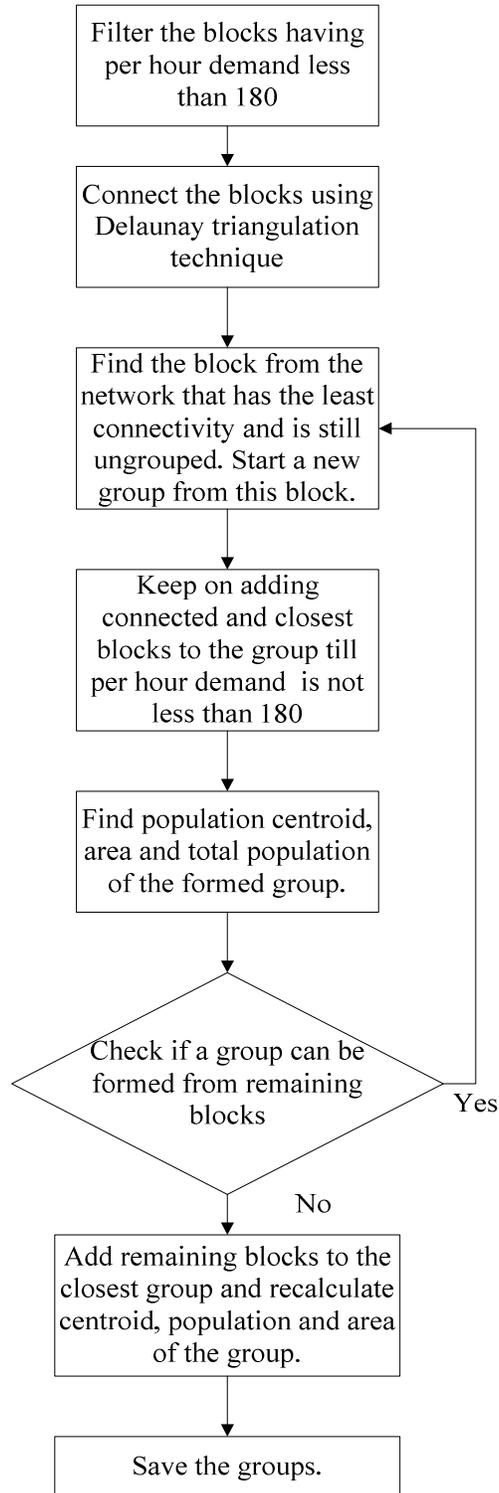


Figure 3.6 Flowchart for Group Formation

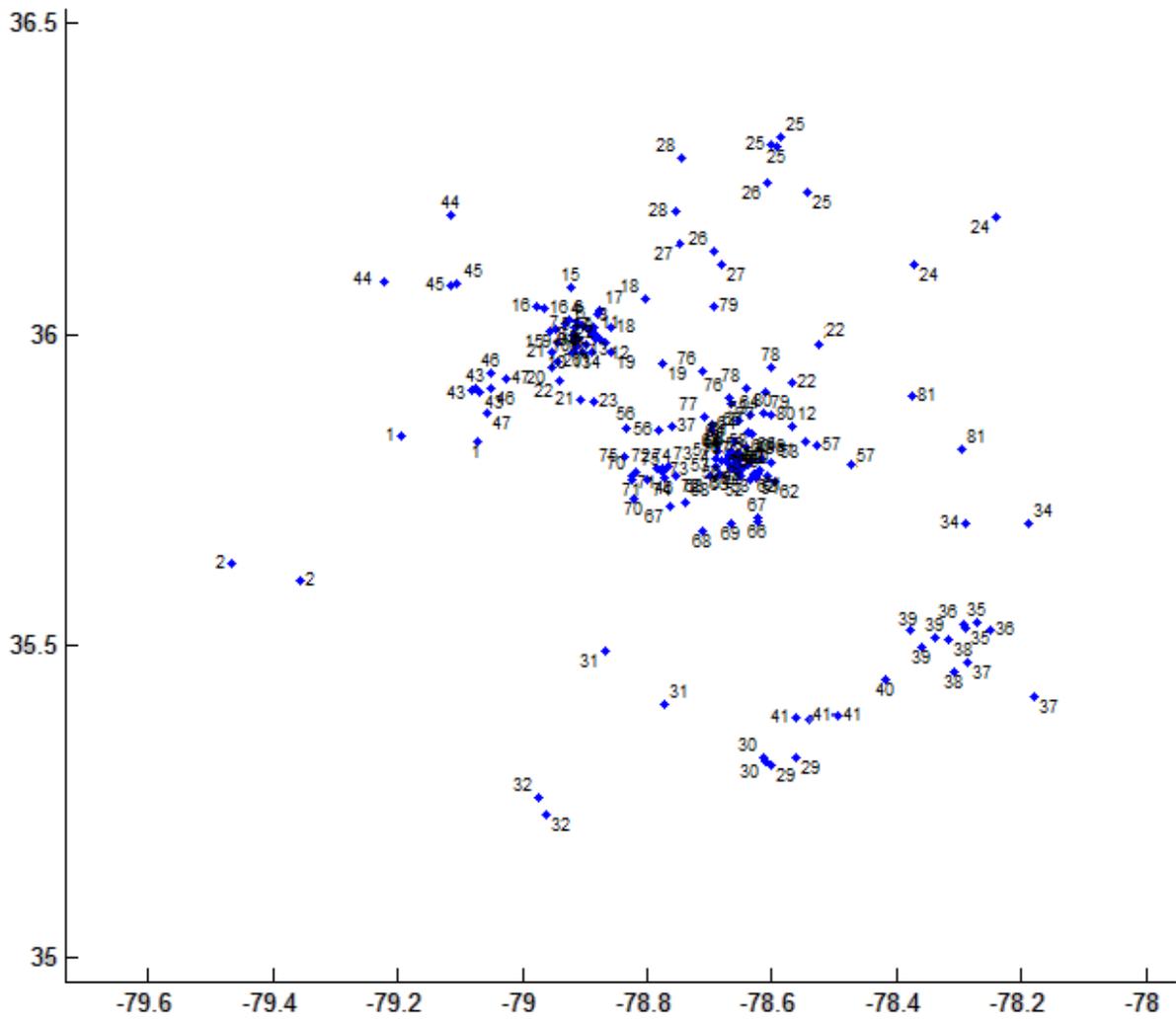
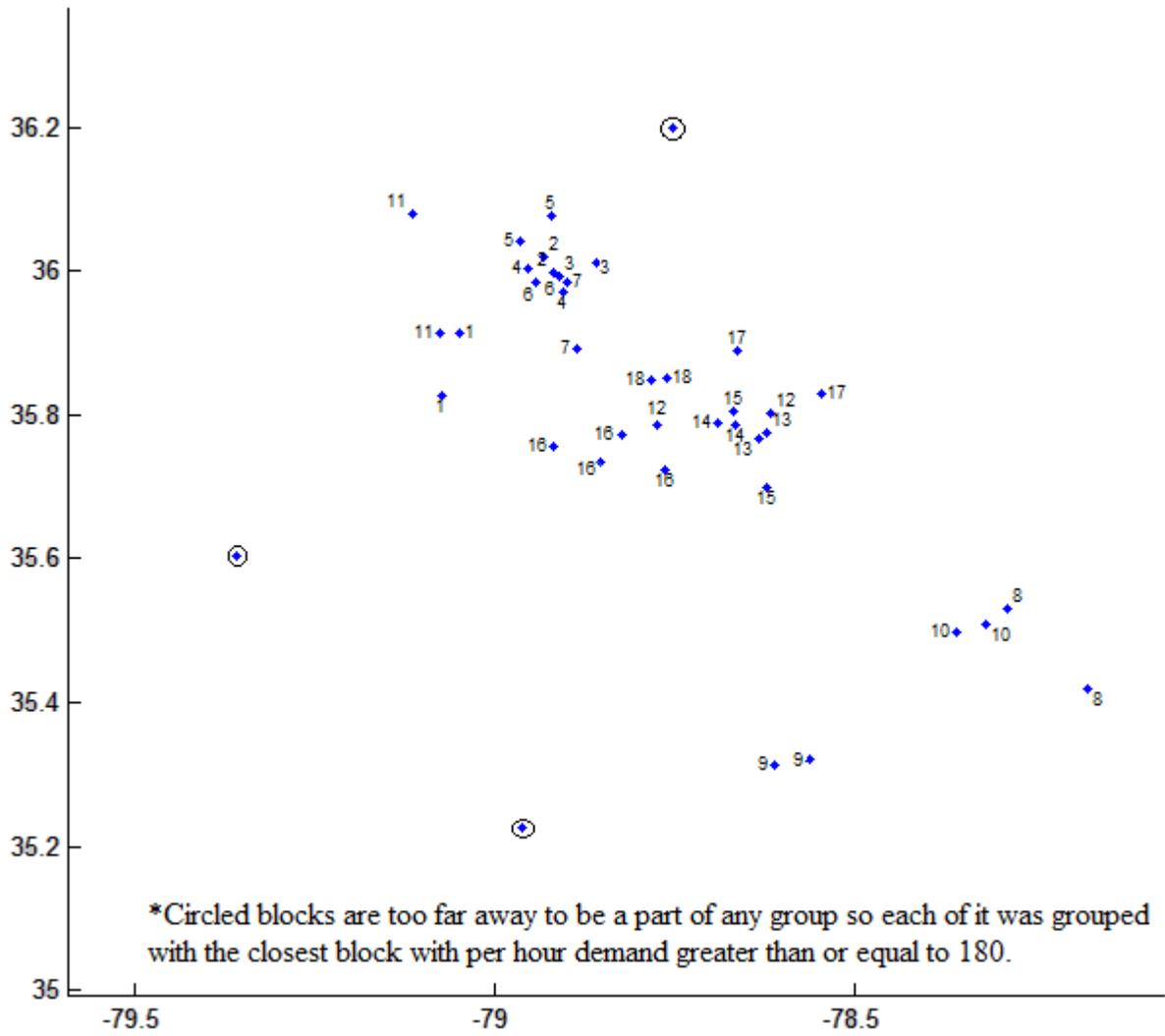


Figure 3.7 Group Formation for $pcd = 2.29$



3.4.3 Estimation of Vehicle Route Distance for Census Block and Group of Census Blocks

The continuum approximation method explained in section 3.4.1 for estimation of the approximate route distance can be used to design vehicle routes for a census block and a group of census blocks. Figure 3.9 shows vehicle routes for a block and a group.

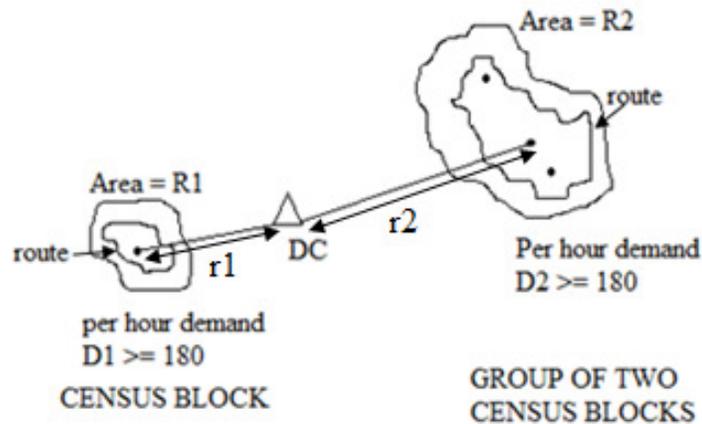


Figure 3.9 Estimation of Vehicle Route Distance

Allocation of census blocks and groups to the nearest DC in the PLN will form clusters such that each cluster will have only one DC and all the blocks and groups in that cluster will be allocated to that DC. Above figure shows a block of area $R1$ and per hour demand $D1$ and a group of two blocks of area $R2$ and per hour demand $D2$ are allocated to a DC.

Line-haul distance for a block = $r1$

Line-haul distance for a group = $r2$

The package carrying capacity of a truck is assumed to be varying between 180 and 240 as per estimated in section 2.6. Therefore, number of packages delivered by a truck per trip is $MIN(D1, 240)$ for a block and $MIN(D2, 240)$ for a group. Suppose $D1 = 480$, then a truck will deliver 240 packages per trip while number of trips will be 2 per hour or 28 per day. For $D1 = 190$, truck will deliver 190 packages per trip while number of trips will be 1 per hour or 14 per day. It is assumed that the number of customers C visited by a truck in a block or in a group is equal to number of packages carried by a truck. Therefore, $C = N = MIN(D1, 240)$.

Weighted line-haul distance for a block = $r1 \times \min(D1, 240)$

Weighted line-haul distance for a group = $r2 \times \min(D2, 240)$

The distance traveled by the truck inside the block = $kG\sqrt{\min(D1, 240) \times R1}$

The weighted distance traveled by a truck inside the block

$$= (kG\sqrt{\min(D1, 240) \times R1}) \times \frac{\min(D1, 240)}{2},$$

where, k = dimensionless proportionality constant for Euclidean metric 0.57,

G = circuitry factor = 1.5

Weighted tour distance for block

$$= (r1 \times \min(D1, 240)) + (kG\sqrt{\min(D1, 240) \times R1}) \times \frac{\min(D1, 240)}{2}$$

Similarly,

Weighted tour distance for group

$$= (r2 \times \min(D2, 240)) + (kG \sqrt{\min(D2, 240) \times R2}) \times \frac{\min(D2, 240)}{2}$$

The waiting time of a package for delivery truck in case of both block and group is,

$$\text{Weighted waiting time for truck} = \left(\frac{14}{\text{trips / day}} \times \text{HeadwayRatio} \right) \times \min(D, 240)$$

where, D = per hour demand (of block/group)

$$\text{trips/day} = \frac{\text{demand / day}}{240} \text{ for } D > 240$$

$$\text{trips/day} = 14 \text{ for } 180 \leq D \leq 240.$$

This completes the calculation for estimating the approximate vehicle route distance and waiting time for truck.

While delivering packages to the customers, the unloading time of one package is assumed to be equal to 30 seconds. Suppose there are 4 customers to be served then total weighted unloading time will be $(4 \times 30) + (3 \times 30) + (2 \times 30) + (1 \times 30) = (\text{sum of first 4 natural numbers}) \times 30 = 300$ seconds. Similarly, for N number of packages delivered by a truck in a block or a group, unloading time would be $\frac{N(N+1)}{2} \times 30$ seconds.

A function *contNetCost2.m* is created using pseudo code given in figure 3.4. This function generates the network of DCs and calculates the sum of all package delays to return the cost of the network (average delivery time). The function *contNetCost2.m* uses the functions *VRforClusters.m* and *VRforNonClusters.m* to calculate local distribution delays for groups and for blocks respectively.

```

1. load ('PLN'); load ('DTP'); load ('RTAblockinfo'); load ('blockarea');
2. if (pcd==2.29)
load ('clusarea229'); load ('cluspop229'); load ('clusidx229'); load ('clusblk229'); end
3. if(pcd==3.48)
load ('clusarea348'); load ('cluspop348'); load ('clusidx348'); load ('clusblk348'); end
4. if (pcd==5.28)
load ('clusarea528'); load ('cluspop528'); load ('clusidx528'); load ('clusblk528'); end
5. WT = 0; insideblk = 0;           % waiting time = 0; inside block travel distance = 0
6. idx = indices of clusters allocated to DC
   If idx == empty
       routedist = 0; WT = 0;
   else
r = Dists (clusters, DC, 'miles') * (daily clusters' demand) / (daily total demand)
for (all clusters in idx)
   if (per hour demand < 240)
       time = HWratio * (daily cluster demand) / (daily total demand);
       WT = WT + time;
inb = (k*G * sqrt (clusarea (idx,1) * per hour demand)) / (vehicle speed);
inb = (inb * (daily demand of cluster)) / (2 * total daily demand);
insideblk = insideblk + inb;
   end
   if (per hour demand >= 240)
       time = HWratio * ( 14 / ((daily cluster demand) / 240));
       time = time * (daily cluster demand) / (daily total demand);
       WT = WT + time;
inb = (k*G * sqrt (clusarea (idx,1) * 240)) / (vehicle speed);
inb = (inb * (daily demand of cluster)) / (2 * total daily demand);
insideblk = insideblk + inb;
   end
end
routedist = r + insideblk;

```

Figure 3.10 Pseudo Code for Estimating Routing Delays for Clusters/Groups

```

1. load ('PLN'); load ('DTP'); load ('RTAblockinfo'); load ('blockarea');
2. WT = 0; insideblk = 0;           % waiting time = 0; inside block travel distance = 0
3. idx = indices of blocks allocated to DC
   if idx == empty
       routedist = 0; WT = 0;
   else
       r = Dists (blocks, DC, 'miles') * (daily group's demand) / (daily total demand)
   for (all blocks in idx)
       if (per hour demand < 240)
           time = HWratio * (daily block demand) / (daily total demand);
           WT = WT + time;
           inb = (k*G * sqrt(blockarea (idx,1) * per hour demand)) / (vehicle speed);
           inb = (inb * (daily demand of block)) / (2 * total daily demand);
           insideblk = insideblk + inb;
       end
       if (per hour demand >= 240)
           time = HWratio * ( 14 / ((daily block demand) / 240));
           time = time * (daily block demand) / (daily total demand);
           WT = WT + time;
           inb = (k*G * sqrt (blockarea (idx,1) * 240)) / (vehicle speed);
           inb = (inb * (daily demand of block)) / (2 * total daily demand);
           insideblk = insideblk + inb;
       end
   end
   routedist = r + insideblk;

```

Figure 3.11 Pseudo Code for Estimating Routing Delays for Blocks

3.4.4 Demonstration of Design of Vehicle Routes

Following figure shows allocation of a census block and a group of two census blocks to the closest DC in the vicinity.

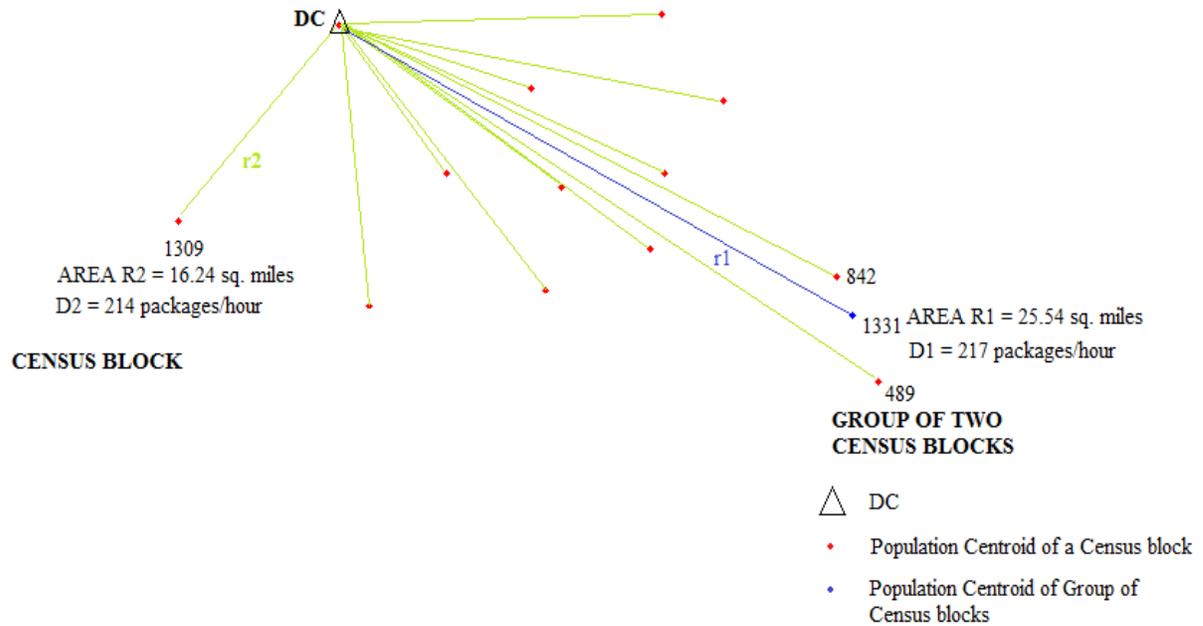


Figure 3.12 Demonstration of Design of Vehicle Route

In above figure, observe the two census blocks having population of 842 and 489. These blocks will not generate per hour package demand more than 180 when daily per capita demand (pcd) is equal to 2.29. During preprocessing stage, these two census blocks are combined to form a group with total population of 1331. The area of the group is $R1 = 25.54$ sq. miles which is the sum of individual areas of the census blocks forming the group. The package demand per hour generated by this group will be equal to $D1 = 217$. The location of the population centroid of the group is calculated in the same way a center of mass or center

of gravity of a system is calculated. The distance $r1$ between DC and population centroid of the group is the great circle distance in miles calculated by assuming the circuitry factor equal to 1.5. The distance between DC and centroid of the census block is $r2 = 3.54$ mi whereas the distance between DC and centroid of group is $r1 = 8.01$ mi.

As discussed in section 3.4.3, the approximate vehicle route distance for census block can be estimated as given below:

$$\text{Weighted line-haul distance for block} = r2 \times \min(D2, 240) = 3.54 \times 214 = 757.56 \text{ mi}$$

The weighted distance traveled by a truck inside the block

$$= (kG \sqrt{\min(D2, 240) \times R2}) \times \frac{\min(D1, 240)}{2}$$

$$= 0.57 \times 1.5 \times \sqrt{214 \times 16.24} \times \frac{214}{2}$$

$$= 5393.24 \text{ mi}$$

$$\text{Thus weighted tour distance for block} = 757.56 + 5393.24 = 6150.80 \text{ mi.}$$

In similar way, the weighted tour distance for group can be calculated by using following equation.

Weighted tour distance for group

$$= (r1 \times \min(D1, 240)) + (kG \sqrt{\min(D1, 240) \times R1}) \times \frac{\min(D1, 240)}{2}$$

Weighted tour distance for group

$$= (8.01 \times 217) + (0.57 \times 1.5 \times \sqrt{217 \times 25.54}) \times \frac{217}{2}$$

$$= 1738.17 + 6906.15 = 8644.32 \text{ mi.}$$

Since the hourly package demands generated by both, block and group, lie between 180 and 240, there will be total 14 trips per day from DC to block and group each.

$$\text{Therefore, weighted waiting time for truck} = \left(\frac{14}{\text{trips / day}} \times \text{HeadwayRatio} \right) \times \min(D, 240)$$

$$= \left(\frac{14}{14} \times 1.7 \right) \times \min(D, 240) = 1.7 \times \min(D, 240)$$

$$\text{Weighted waiting time for truck inbound to the block} = 1.7 \times 214 = 363.8 \text{ hr.}$$

$$\text{Similarly, weighted waiting time for truck inbound to the group} = 1.7 \times 217 = 368.9 \text{ hr.}$$

Additionally; weighted unloading time of packages inbound to the block would be equal to $\frac{214(214+1)}{2} \times 30 = 690150 \text{ sec} = 191.70 \text{ hr}$ whereas weighted unloading time of packages inbound to the group would be equal to 197.01 hr.

3.5 Search Heuristics

For a given network of DCs in the RDC, the average package delivery time is,

$$\text{Average delivery Time} = \text{waiting time} + \text{loading-unloading time} + \text{DC-to-DC transportation time} + \text{local transportation time}$$

In this section, a search procedure is developed that minimizes the delivery time by forming a network for a given number of DCs. Since it is unknown how many DCs should be located, the search procedure is run iteratively by incrementally increasing the number of DCs. The iteration, which gives the minimum value for average delivery time, is considered as the near-optimal solution.

A variety of heuristics have been developed for the facility location and allocation problem. One simple and accurate method is the destination subset algorithm [9]. But this procedure is computationally heavy since it checks a total of $n!/m!(n-m)!$ combinations for finding m DC locations from n possible locations. Leon Cooper [9] introduced the alternate location-allocation (ALA) method that is efficient and quick. Based on Cooper's method, Kay [11] developed the ALA procedure in Matlab that uses gradient based search and Nelder-Mead search to locate the DCs. This procedure searches for best DC locations across a set of possible location in the map. In the PLN design, only those locations which are on the network are considered as potential locations. Bucci [10] developed a discrete ALA procedure for solving a discrete multi-facility location problem. Bansal [2] used a genetic algorithm for facility location which is used for solving combinatorial optimization problems.

In this research, a search heuristic is developed which is inspired by the discrete ALA procedure introduced by Bucci [10]. This method performs the neighborhood search to find the improved locations, which is then followed by the allocation process. New locations are found by the neighborhood search and the procedure is repeated until no improvement is observed. To increase effectiveness, weak points from the network are eliminated from consideration as potential DC locations and a starting set of DCs is sampled from remaining points.

3.5.1 Creating the Set of Potential Points

A point on the network can either be a census block or road intersection. The census blocks are demand/destination points while the DCs will be supply/source points. As per the random destination algorithm [9], the destination set is the favored set from the point of view of locating sources. Since all the blocks in a cluster will be allocated to the same DC, the most probable best location for that DC will be at one of the blocks/destinations. This is also justified by the fact that the probability that a point will be a possible best location for DC is directly proportional to the population surrounded by that point. Since the census blocks are the points where population of that block is concentrated, blocks can be considered as better locations for DCs than other points in the network.

In the final URN, there are a total of 1384 points including 925 census blocks. Hence, a search for the DC locations can be narrowed to only 925 points by eliminating 459 points which are not census blocks. Now suppose n DCs must be located. Then there can be a total

of $\frac{925!}{n!(925-n)!}$ combinations, which is a very large number. So, 925 census blocks were filtered again by using a weighted binary selection method developed by Kay in the Matlog toolbox [11] (*wbinselect.m*). The weight of a given block is the population associated with that block. This selection method generates n random numbers $r_1, r_2, r_3, \dots, r_n$ between 0 to 1 where n is total number of points in consideration i.e. 925. Numbers a_1, a_2, \dots, a_n are calculated by dividing the weight of each block with average weight. A block i is chosen if $r_i < pa_i$ for all $i = 1, \dots, n$ where p is a parameter called probability of choosing any node depending on its weight. Using this procedure, a total of 201 blocks were selected and saved by name *IniGuess.mat*.

If n near-optimal DC locations are required to find then a function *BestSol.m* randomly samples n blocks from *IniGuess.mat* followed by generation of the network of DCs and calculation of the cost (average package delivery time) for the network of DCs. Cost is calculated for 100 different random combinations of the n blocks and the combination with the minimum cost is saved by name *StartingDCs.mat* as starting point for search heuristic. Note that, this procedure helps in finding better starting point combination from the set of 201 potential blocks, but the search heuristic will take all census block locations into consideration while searching for a near-optimal solution.

3.5.2 ALA Using Neighborhood Search

The search heuristic can be illustrated using an example. Suppose 12 DCs must be located in a PLN. Using the function *BestSol.m*, best combination of 12 DCs is found from the set of

201 blocks as a starting guess. Total cost of the network is estimated to be equal to 155.6 minutes. Block allocation and DC locations are shown in Figure 3.12.

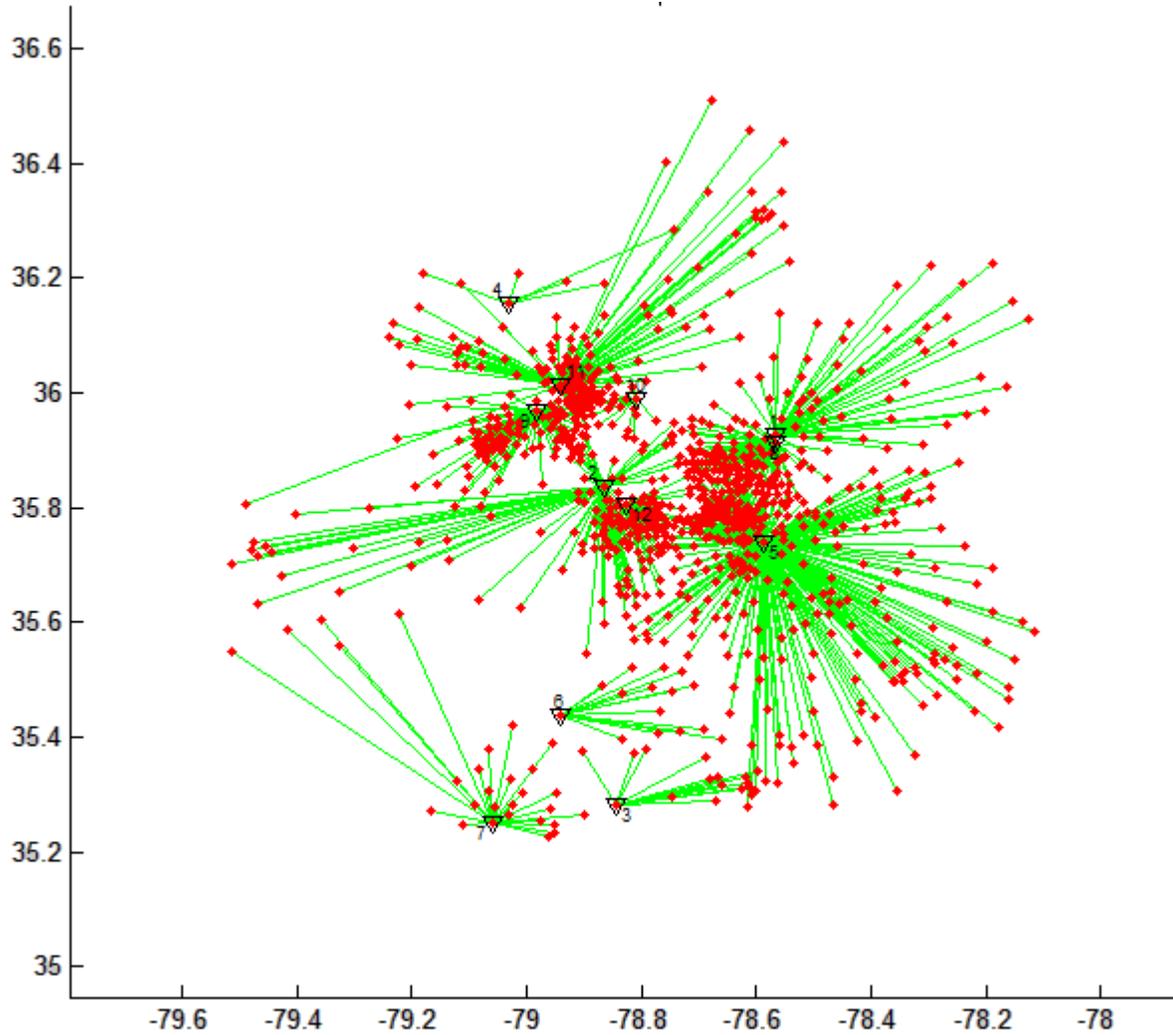


Figure 3.12 Starting Locations and Allocations of 12 DCs

The parameters used to calculate the network cost are per-capita daily package demand of 5.28, a loading-unloading time of 5 minutes, 80% load factor for a truck with capacity of 300 packages and a headway ratio equal to 1.7.

Since the DCs are located and allocations are known, new improved locations need to be found. To find new locations, the neighborhood search is performed in each cluster starting from cluster 1 to cluster 12. Cluster 1 denotes a cluster of blocks in which all blocks are allocated to DC1. DC1 is relocated from its original location to every possible location in cluster 1 and each time a new cost of the modified network is estimated. The location where the cost of the network is the minimum, DC is re-located to that location until the next iteration. While finding the location for DC i in a cluster i , the locations of all other DCs will be kept fixed. The same procedure repeated in all 12 clusters to find new locations for all the DCs. Thus during this process, the cost of the network is calculated 925 times. Improved locations are shown in figure 3.13. Total cost of this network is 146.57 minutes. This concludes the first iteration with the completion of location procedure.

It is easy to find the allocation of the blocks to the DCs according to their new improved locations. Now since allocations are changed, all clusters are also changed. Hence the same neighborhood search as explained earlier is repeated in each new cluster to find another set of locations for 12 DCs. In a similar way, this location and allocation procedure is repeated until no reduction in cost is found. At this point, the search procedure stops and the locations from the last iteration will be the DCs' final locations which are shown in Figure 3.14. The cost of this near-optimal network is 142.9 minutes. Heuristic can be re-run for the different number of DCs for a given set of parameter values. It is found that the average delivery time is the least when 41 DCs are located. Solution shown in Figure 3.13 is the solution for given set of parameter values when number of DCs in the PLN is 12.

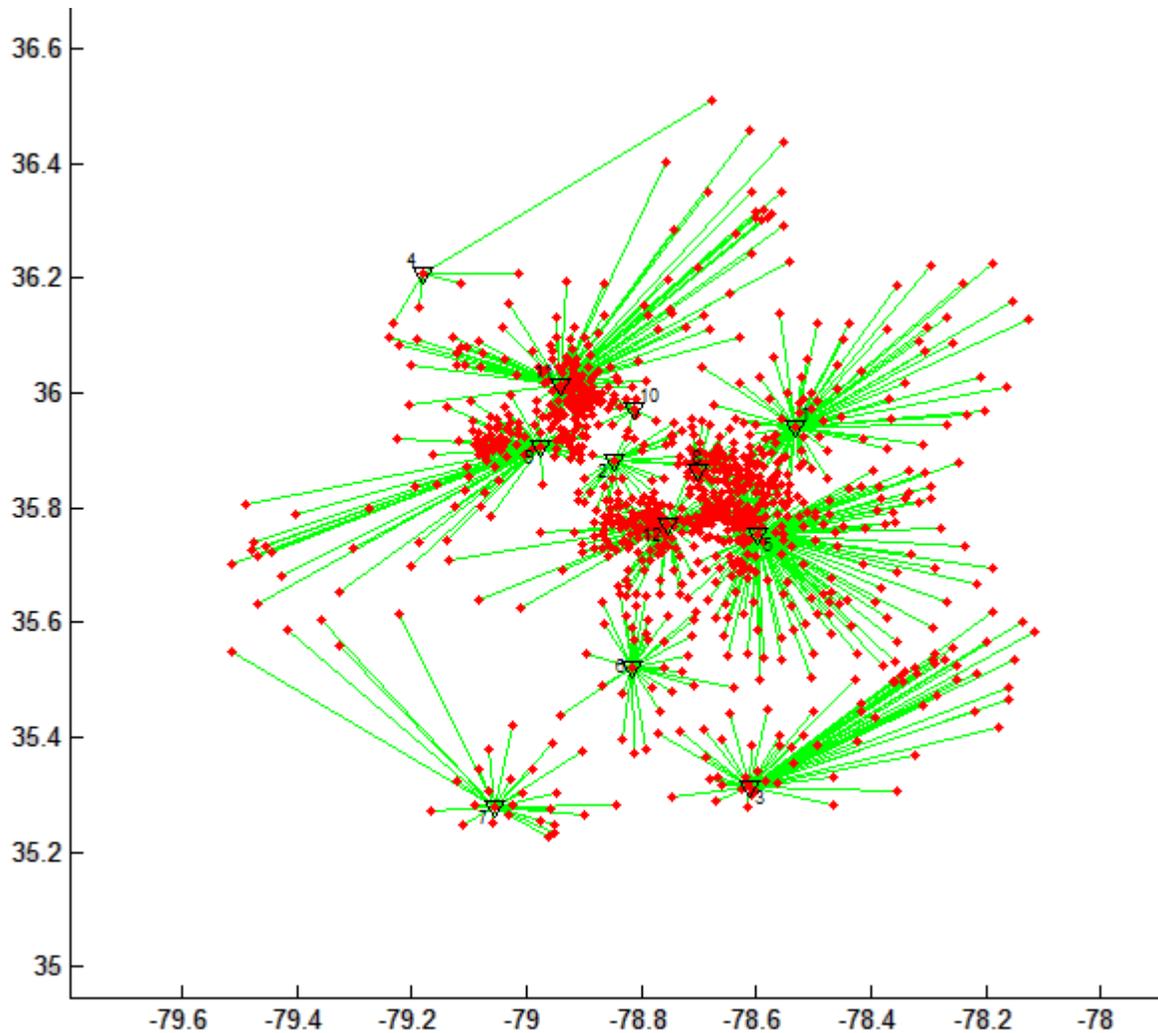


Figure 3.13 Improved DC Locations and Allocations after 1st Iteration

The near-optimal solutions are found for different values of per-capita daily package demand in the range [2.29, 5.28] and different loading-unloading times in the range of [0, 10]. Also for given values of parameters, different types of package delays are calculated when number of DCs is increasing in the PLN. In the next section, trends and patterns of these delays are studied.

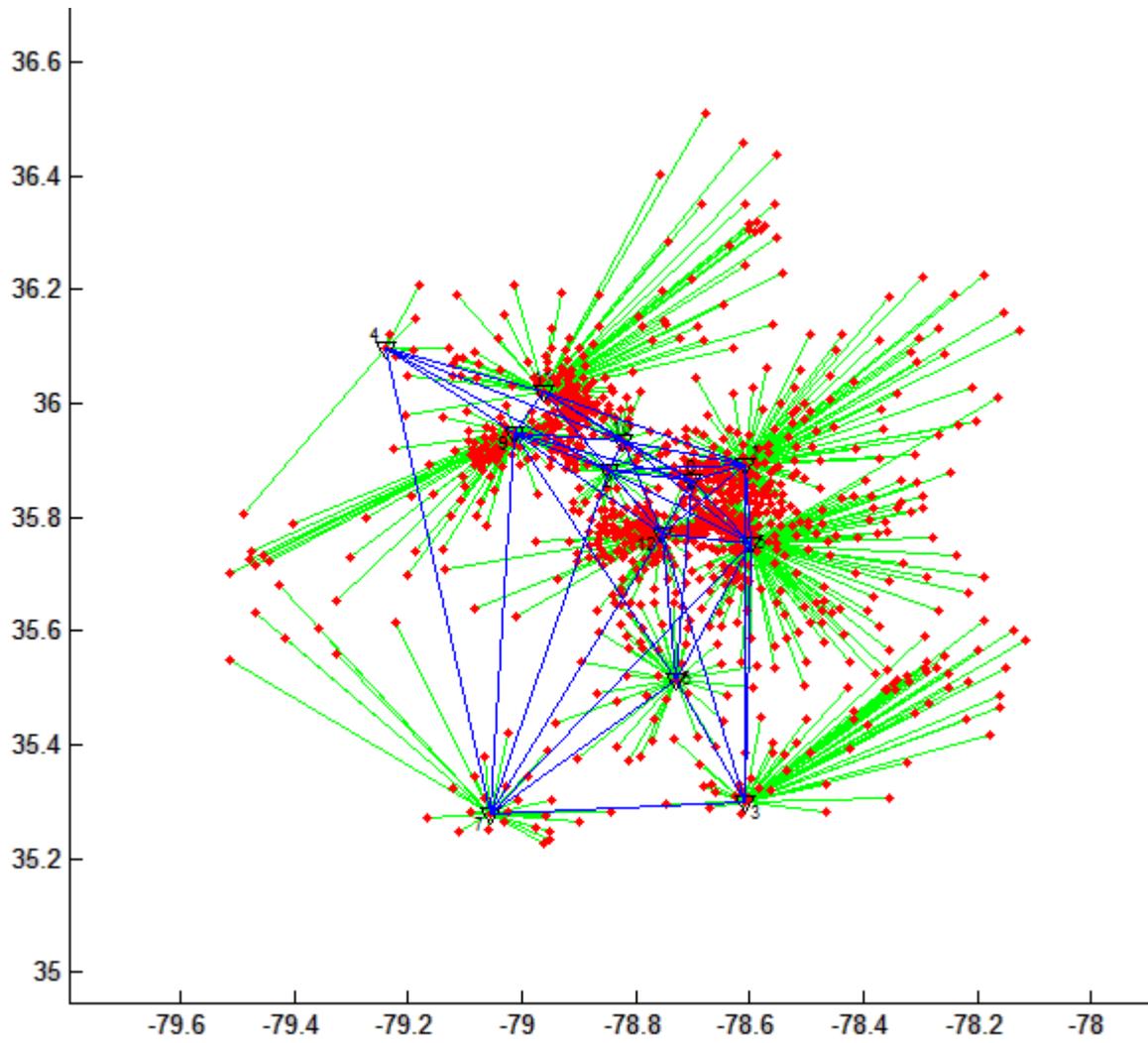


Figure 3.14 Best Locations and Network of DCs

The pseudo-code for heuristic *NSALA.m* is shown in figure 3.15.

```
1. Load starting  $n$  DCs saved for given set of parameters in data file StartingDCs.mat.
2. Load (URN, shortest time T, paths P and distances D and areas of census blocks).
3. TC = Average delivery time for starting  $n$  DCs calculated using contNetCost2.m.
4. optTC = TC; optDC = DC; done = 0; % DC and optDC variables stores starting  $n$  DCs.
6. while (~done)
    row = 1; % value "1" indicates neighborhood search will be started with first DC.
    while (row <= number of DCs i.e.  $n$ )
        find blocks allocated to rowth DC.
        L = 1; % "L" is index of allocated block to rowth DC.
        while (L <= total number of allocated blocks to rowth DC)
            tempDC = optDC;
            rowth DC in tempDC = Lth block allocated to rowth DC;
            tempTC = average delivery time for tempDC using contNetCost2.m
            optTC = min (optTC, tempTC);
            if (optTC == tempTC)
                optDC = tempDC;
            end
            L = L+1;
        end
    row = row+1;
    end
if ( TC – optTC >= 0.5 )
    TC = Average delivery time for starting optDC calculated using contNetCost2.m; done = 0;
    optTC = TC;
else done = 1; end end
```

Figure 3.15 Pseudo-Code for Heuristic

Chapter 4: Results, Conclusion and Future Work

4.1 Results

The results are collected for different sets of parameters. The set of parameters include loading-unloading time and per-capita demand value. There are three values of loading-unloading time which are taken into consideration: 0 minutes, 5 minutes and 10 minutes, while per capita demand (pcd) varies between [2.29, 5.28] with the most likely value of 3.48. The following procedure is used to find the best locations of n DCs using the neighborhood search heuristic mentioned in section 3.5.2. First, the starting n locations should be found from the *IniGuess.mat*. This can be done by using function *BestSol.m* as explained in 3.5.1 and the starting locations can be saved by name *StartingDCs.mat* for varying values of n and a fixed set of parameters. For each n value there is a set of locations saved in *StartingDCs.mat* which is fed as an input to the neighborhood search process to find best n locations. The value of n for which the average delivery time is the minimum is the best solution for that set of parameter values.

4.1.1 Best Case Scenario

First set of parameter values is: per-capita demand (pcd) = 5.28 and loading-unloading time (LU) = 0 minutes (Best Case). Value of n is increased from 20 with increment of 5. For each n value near-optimal DC locations and the minimum average delivery times are found. It was observed that as n increases, average delivery time decreases. Average delivery time was the

minimum when $n = 40$ while from $n = 45$, it started increasing. Thus search for near-optimal solution was executed between $n = 40$ and $n = 45$ with increment of 2.

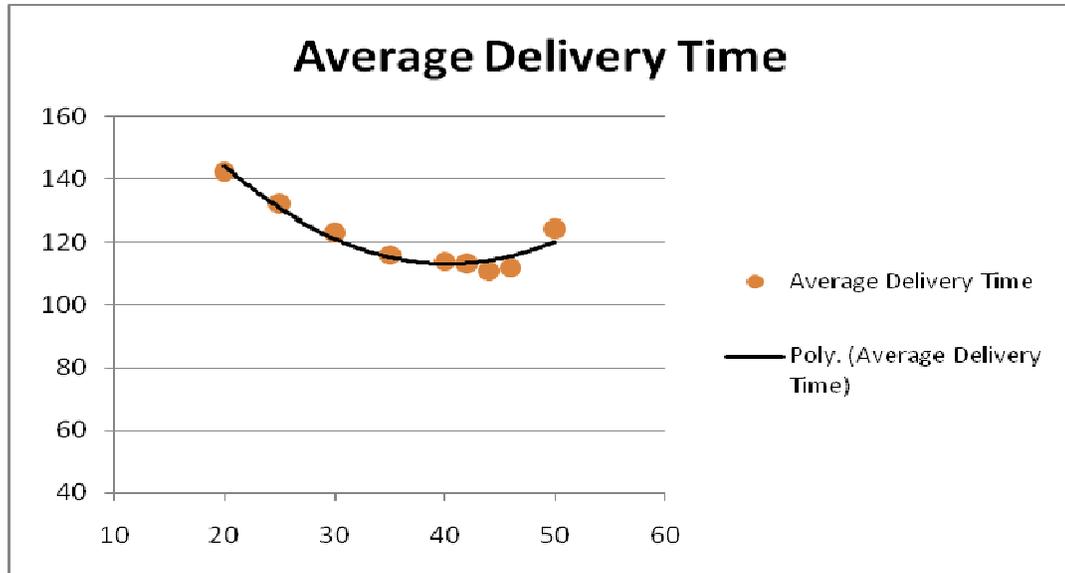


Figure 4.1 Delivery Time vs. Number of DCs for $pcd = 5.28$ and $LU = 0$

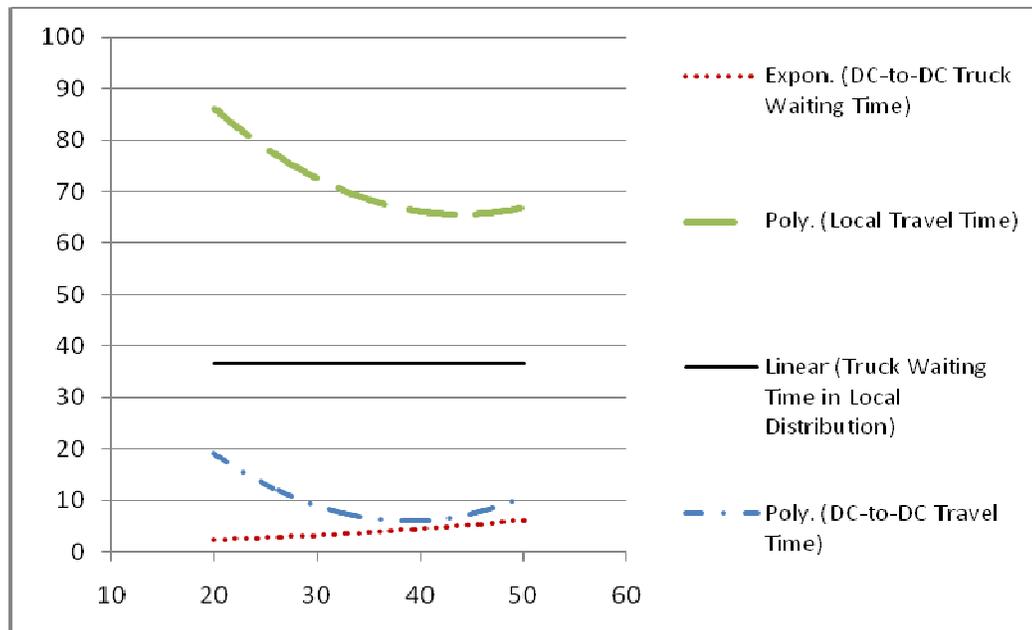


Figure 4.2 Delivery Delays vs. Number of DCs for $pcd = 5.28$ and $LU = 0$

From the results it is observed that for this set of parameter values, the near-optimal solution is obtained at $n = 44$ with the minimum average delivery time = 112.505 minutes. Average delivery times for the near-optimal network of n DCs are plotted against n , i.e. the number of DCs in Figure 4.1. Also, all kinds of delays in the network of near-optimal n DC locations are plotted against n in figure 4.2.

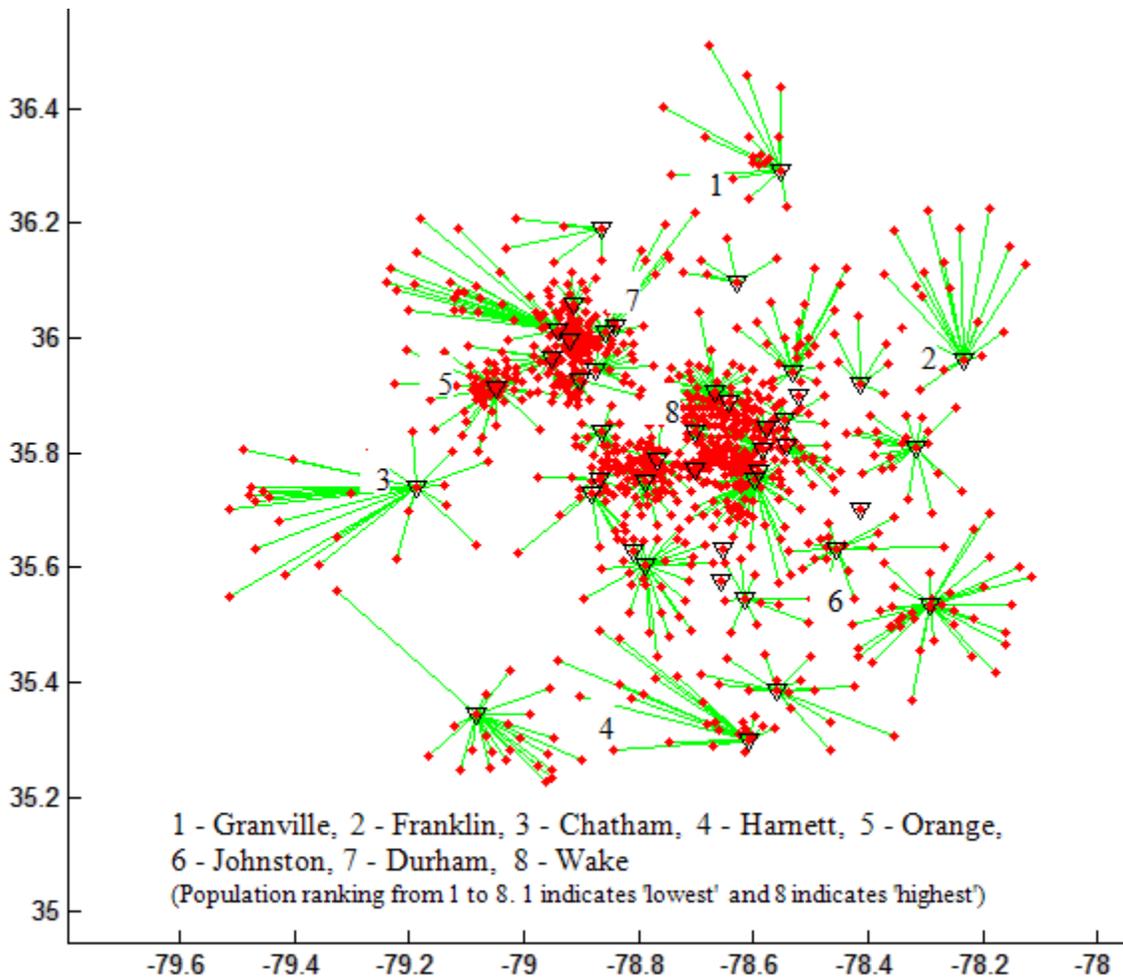


Figure 4.3 Best DC Locations for $pcd = 5.28$ and $LU = 0$ (Best Case)

From Figure 4.3 and Table 2.2, it is clear that the greater is the population of an area, the more DCs will be located in that area. Counties from 1 to 4 are allocated minimum 1 and maximum 2 DCs. County 7 (Durham) shares some of the DCs with the county 5 (Orange). Rest of the DCs are allocated to counties 6, 7 and 8 which carry almost 75% of total population.

The loading-unloading time (LU) is assumed zero in this case and hence not shown in Figure 4.2. For a given value of pcd , demand arising from blocks per-hour will be constant regardless of number and location of the DCs. Hence the number of trips to the blocks and the groups of blocks remains constant. Therefore, the waiting time for a delivery truck in the local distribution is constant for all values of n (number of DCs) in the PLN. In this case, it is equal to 36.52 minutes. As the number of DCs increases in the PLN, DC-to-DC travel time decreases initially because the average distance between any two DCs in the network decreases. After certain value of n , the DC-to-DC travel time increases with increase in n . When n was less, packages could be transported from one DC to the destination DC via less number of DCs on their way. With an increase in the number of DCs in the PLN, the average number of DCs visited by a package to reach to the destination DC also increases. Because of that, DC-to-DC travel time increases after certain value of n . Furthermore, increase in the n increases the truck headway between DCs resulting into the increase in the DC-to-DC truck waiting time. It is logical that with an increase in n , the local travel time from the DC to the customer location decreases. The local travel time includes the package unloading time delay of 30 seconds as well. Because of these trends in different types of delays, the plot of average

package delivery time, which is addition of all the delays against number of DCs is U-shaped curve. Basically, the search heuristic finds trade-offs among DC-to-DC travel time, DC-to-DC truck waiting time, and local travel time. Similar patterns are observed in these delays for all tested values of pcd and LU .

4.1.2 Effect of Loading-Unloading Time on Best Solution

After assuming loading-unloading time $LU = 10$ minutes and $pcd = 5.28$, best solution is obtained at $n = 40$ with average package delivery time of 128.2227 minutes.



Figure 4.4 Delivery Time vs. Number of DCs for $pcd = 5.28$ and $LU = 10$

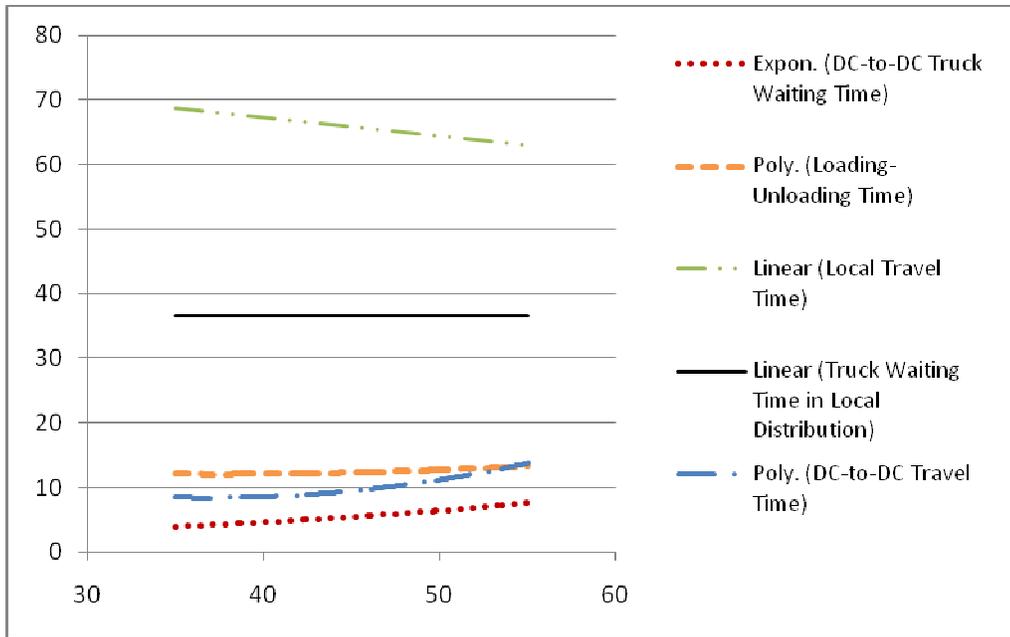


Figure 4.5 Delivery Delays vs. Number of DCs for $pcd = 5.28$ and $LU = 10$

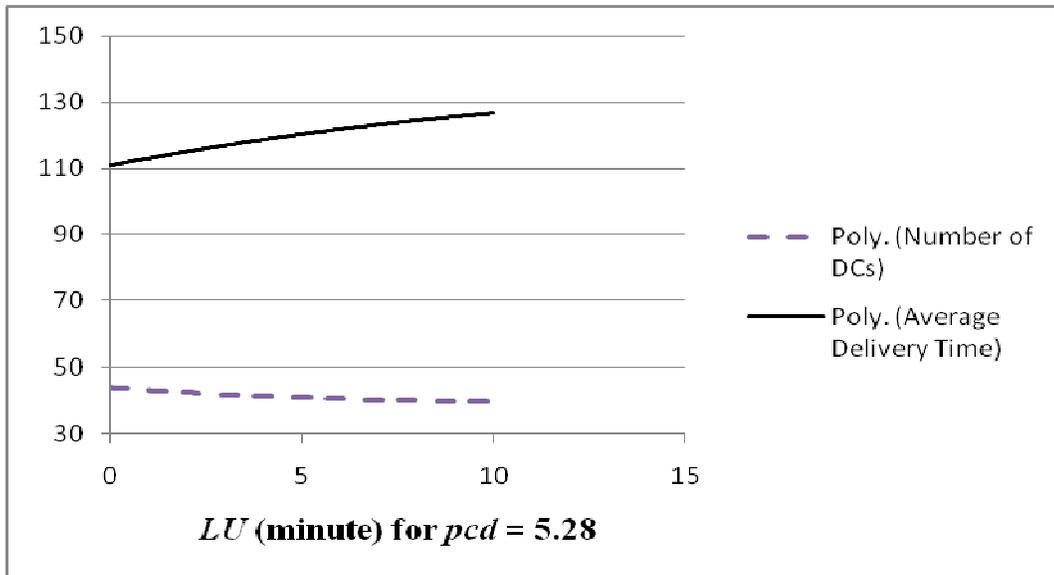


Figure 4.6 LU vs. Optimal Number of DCs and Average Delivery Time

Average loading-unloading time increases with increase in the number of DCs in the PLN because it increases number of DCs visited by a package on its way before reaching to its destination DC. Due to this reason as LU increases, number of DCs in near-optimal solution decrease. Figure 4.4 and 4.5 show plots of average delivery time vs. number of DCs and package delivery delays vs. number of DCs respectively. The effect of LU and pcd on number of DCs in the near-optimal PLN and on the average delivery time is shown in Figure 4.6 and Table 4.1.

Table 4.1 Near-optimal Number of DCs and Average Delivery Time for Varying LU and pcd

LU	0		5		10	
	DCs	Avg. Delivery Time	DCs	Avg. Delivery Time	DCs	Avg. Delivery Time
2.29	32	155.3275	30	159.6634	30	167.6209
3.48	35	136.7338	32	140.6521	31	147.406
5.28	44	112.505	41	121.8486	40	128.2227

4.1.3 Worst Case Scenario

When $pcd = 2.29$ and $LU = 10$ (Worst Case), only 30 DCs can be located in the PLN with the minimum average delivery time of 167.6209 minutes. Average package delivery time and package delays for different values of n are shown in Figures 4.7 and 4.8. Figure 4.9 shows the DC locations in this case. Since this network contains only 30 DCs (less than as compared to 44 for $pcd = 5.28$ and $LU = 0$ minutes), they are well spread so that they can be shared by two or more counties.

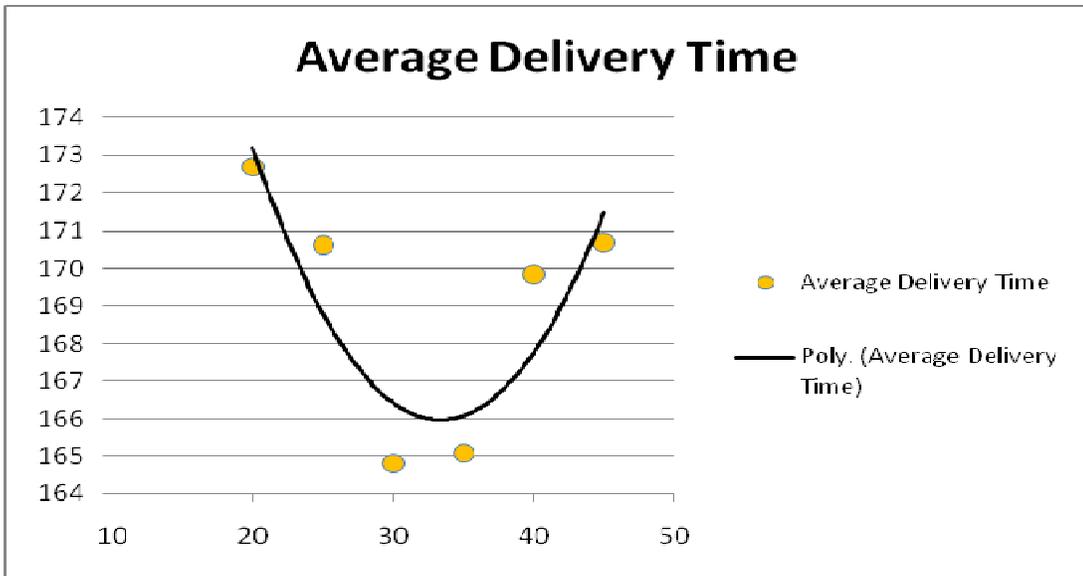


Figure 4.7 Delivery Time vs. Number of DCs for $pcd = 2.29$ and $LU = 10$

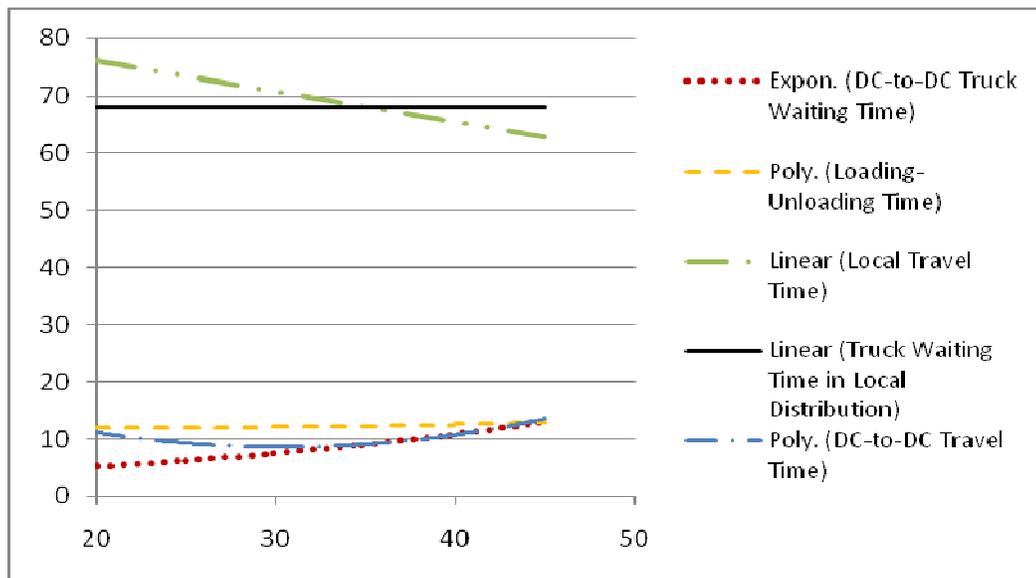


Figure 4.8 Delivery Delays vs. Number of DCs for $pcd = 2.29$ and $LU = 10$

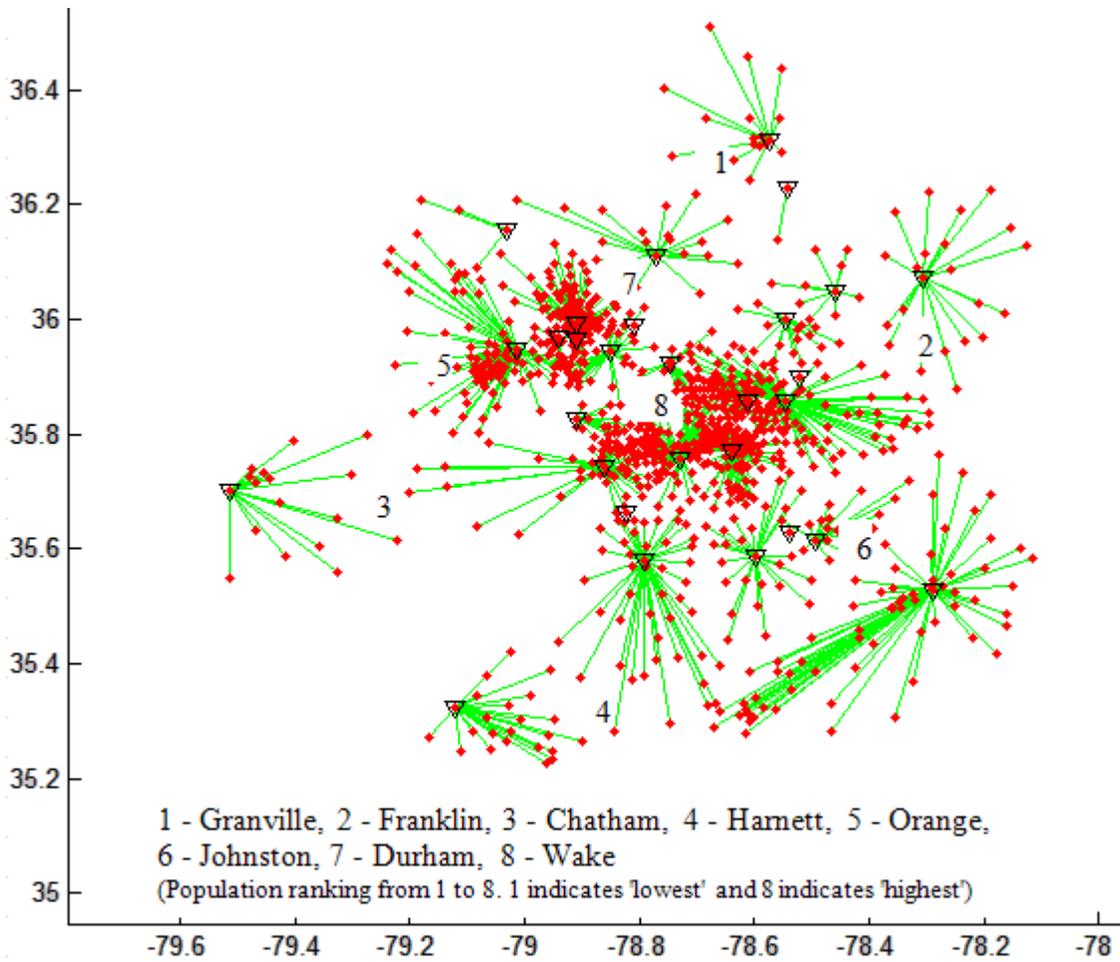


Figure 4.9 Best DC Locations for $pcd = 2.29$ and $LU = 10$ (Worst Case)

4.1.4 Practical Scenario

Geometric mean of the upper (5.28) and the lower (2.29) bounds of per-capita demand pcd is 3.48; the most likely per-capita demand handled by the PLN. For $pcd = 3.48$, the different values of LU in the interval $[0, 10]$ minutes has the same effects on the near-optimal solution and package delivery delays also change in similar manner with number of DCs n as discussed in best and worst case scenarios.

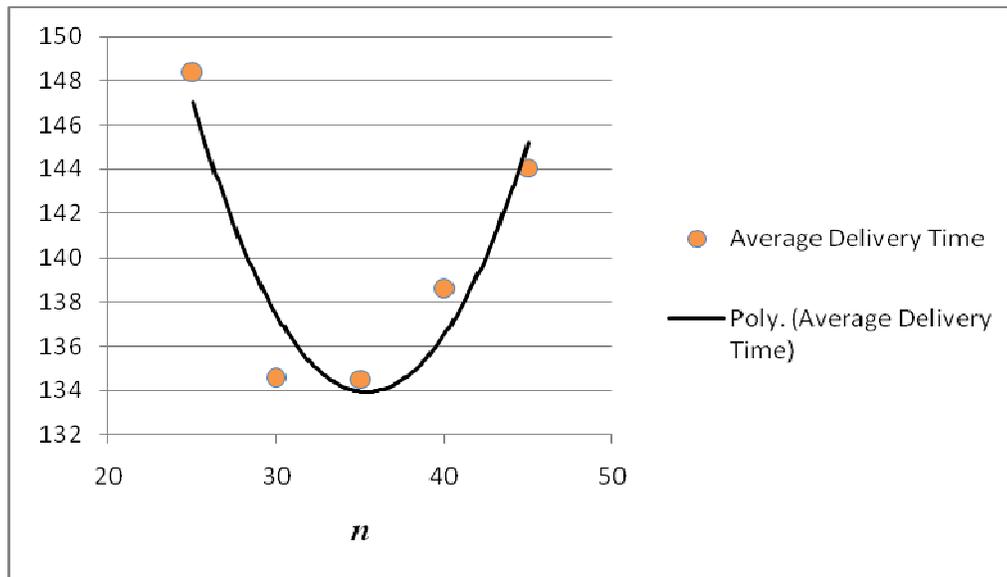
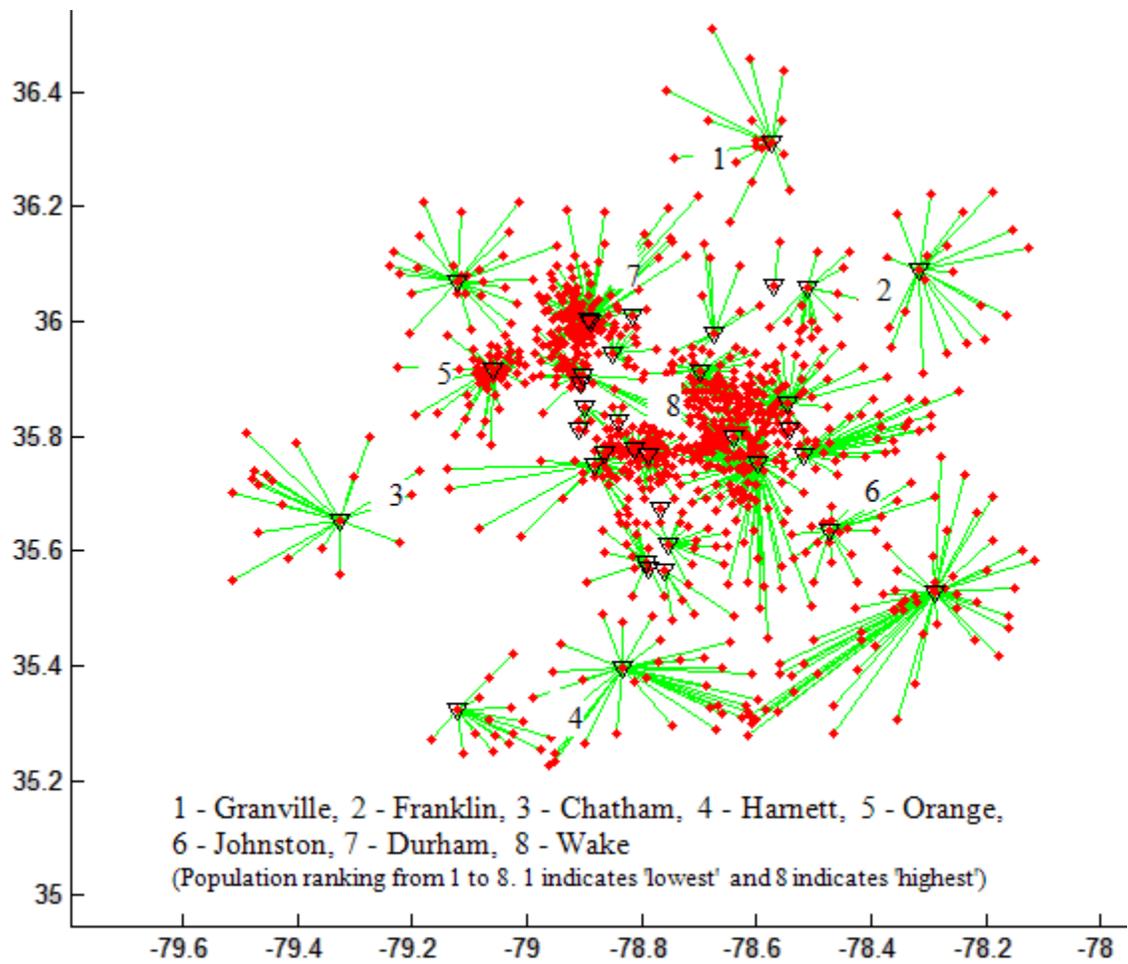


Figure 4.10 Delivery Time vs. Number of DCs for $pcd = 3.48$ and $LU = 0$

Figure 4.10 shows minimum average delivery time values for different number of DCs n when $LU = 0$. The near-optimal solution is obtained when there are 35 DCs in the PLN with the average delivery time of 136.7338. Figure 4.11 shows the DC locations and allocations on map of RDC. Figure 4.12 shows the package delays vs. number of DCs n for $LU = 0$ minute.



**Figure 4.11 Best DC Locations for $pcd = 3.48$ and $LU = 0$
 (Practical Case)**

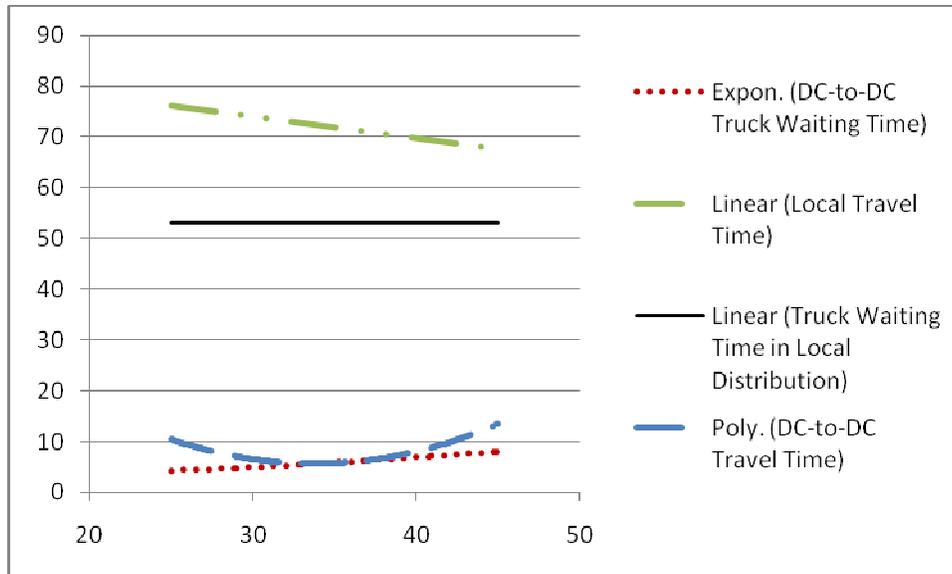


Figure 4.12 Delivery Delays vs. Number of DCs for $pcd = 3.48$ and $LU = 0$

In practical situations, loading-unloading time cannot be 0 minute. If the minimum and the maximum times for loading-unloading are assumed equal to 5 and 10 minutes, respectively then in practical case average delivery time varies between 140.6521 and 147.406 minute. Average delivery time can further be reduced by reducing the time for loading-unloading operation below 5 minutes.

4.1.5 Waiting Time for Truck in DC-to-DC and Local Distribution

From Figures 4.2, 4.5 and 4.12, it is clear that a package is significantly delayed because of the wait for a local distribution delivery truck. As discussed earlier, this delay is almost constant for given value of pcd irrespective of LU and the number of DCs in the PLN. For $pcd = 5.28$ and $pcd = 2.29$, the average waiting times for a delivery truck for local distribution are 36.52 minutes and 68.13 minutes respectively. As pcd decreases, the package demand arising from census blocks also decreases linearly. The decrease in package demand

causes reduction in the number of trips to the blocks which in turn increases the truck headway. The waiting time of a package for local delivery truck is the product of headway ratio (1.7) and truck headway. Hence the average local delivery truck waiting time increases exponentially with the decrease in pcd . When pcd is almost halved from 5.28 to 2.29, the local delivery truck waiting time is almost doubled from 36.52 to 68.13 minutes. Figure 4.13 shows exponential relationship between local delivery truck waiting time and pcd .

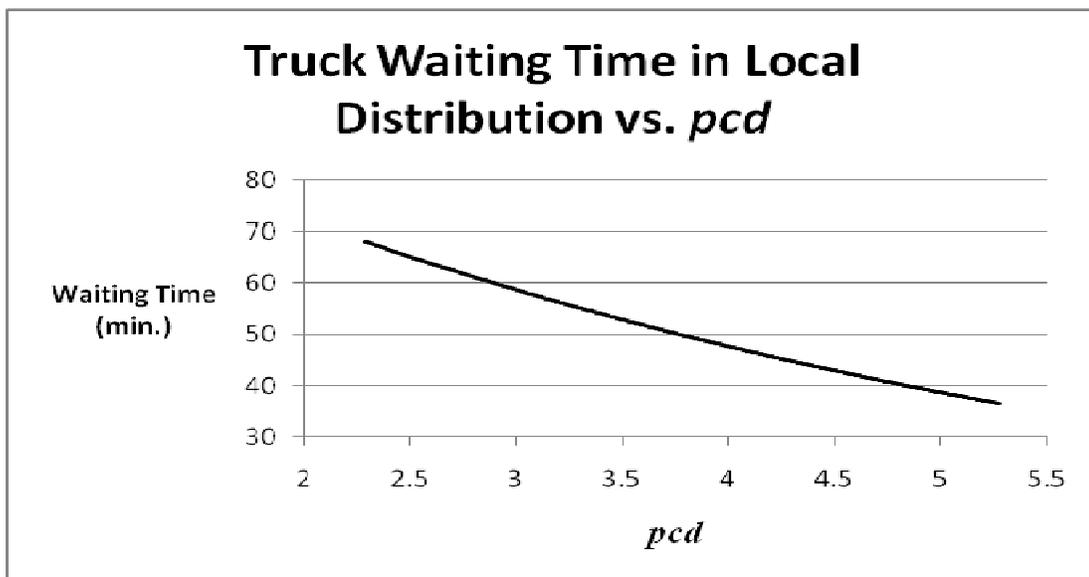


Figure 4.13 Exponential Relation between Local Delivery Truck Waiting Time and pcd

Moreover, it can be inferred from the results that as pcd decreases for a given value of LU , the number of DCs n in the near-optimal PLN also decrease. Comparing figures 4.5 and 4.8, for $LU= 10$ minutes, all delays follow same pattern when plotted against the number of DCs for both values of pcd (5.28 and 2.29) except the DC-to-DC truck waiting time. The rate of increase in the DC-to-DC truck waiting time is more for $pcd = 2.29$ than it is for $pcd = 5.28$.

Hence the DC-to-DC truck waiting time increases slowly in Figure 4.5 with respect to n whereas this same delay increases rapidly in figure 4.8 with respect to n . The change of slope of the DC-to-DC truck waiting time against n with decrease in pcd is the reason for the decrease in number of DCs in the near-optimal PLN as pcd decreases. Figure 4.14 shows the trend between pcd vs. number of DCs and average delivery time for $LU=0$ minute.

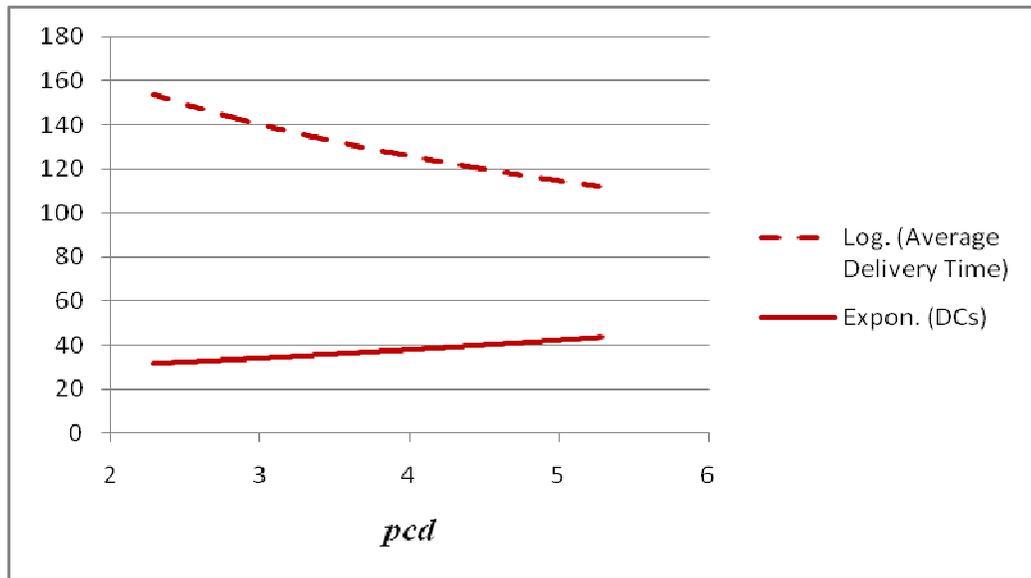


Figure 4.14 Number of DCs and Average Delivery Time vs. pcd for $LU = 0$

4.2 Conclusion

After studying the results for different sets of parameters, following can be concluded:

1. In the best case scenario ($pcd = 5.28$ and $LU = 0$ minutes), the minimum average package delivery time was 112.505 minutes with a total of 44 DCs in the PLN. In the worst case scenario ($pcd = 2.29$ and $LU = 10$ minutes), the minimum average package delivery time is 167.6209 minutes with 30 DCs in the PLN. In the practical case ($pcd = 3.48$, $LU \in [5,10]$), average delivery time varies from 140.6521 and 147.406 minute when LU increases from 5 to 10 minutes while number of DCs decrease from 32 to 31.
2. For pcd varying between $[2.29, 5.28]$ and for any value of LU between $[0, 10]$ minutes, the near-optimal PLN contains between 30 and 44 DCs, and the average package delivery time varies within $[112.505, 167.6209]$ minutes.
3. Local distribution truck waiting time is the most significant delay in the package delivery which is independent of number of DCs in the PLN and LU . This delay increases with decrease in pcd in exponential fashion. The difference in worst case and best case package delivery times is 55.1159 minutes, out of which an increase of approximately 31.61 minutes is observed in the waiting time of package for truck for local distribution.
4. A PLN can feasibly be implemented in RDC for any demand values in the range $[2.29, 5.28]$ for purpose of the local distribution of packages to the customer locations with maximum and minimum delivery time of approximately 2 hour 45 minutes and

2 hour respectively. In most of the cases, the delivery time would vary between 140 to 150 minutes.

4.3 Future Work

- This research estimates upper and lower bounds with the most likely value of per capita demand and generates PLNs for different parameter values. The value of daily per capita demand can be estimated more accurately by studying income of the population in the RDC and their buying capacity.
- While designing the vehicle routes, demand per day is considered to arise over 14 hours. Also, demand per day is assumed to be distributed uniformly over time. In practical situations, demand is not uniformly distributed. An accurate demand distribution over a day can be found to estimate local travel and truck waiting time delays precisely.
- Since demand is assumed to be distributed uniformly over time, the vehicle routing procedure is similar for all time windows. But for a non-uniform distribution, vehicle routes would have to be designed separately for each time window which makes a complete design process computationally heavy. A new analysis method should be developed for route design which is more effective and less time-consuming.
- Inside a census block area, customers are assumed to be distributed uniformly. A detailed population distribution will help to estimate a function which will give population based on XY coordinates in RDC. Using this function, average distance

between two consecutive customers can accurately be estimated. Also, 9-digit ZIP codes will give better resolution of population than the census blocks.

- The performance of the PLN can be studied by studying sensitivity of different parameters using simulation techniques.

REFERENCES

1. Kay, M. G. and Parlikad, A. N., “Material Flow Analysis of Public Logistics Networks”, in *Progress in Material Handling Research: 2002*, R. Meller et al., Eds., Charlotte, NC: The Material Handling Institute, 2002, pp. 205-218.
2. Bansal, A., 2004, Design of A Public Logistics Network, Thesis (M.S.), Dept. of Industrial and System Engineering, North Carolina State University.
3. Xiang, L., 2009, Performance Model for a Public Logistics Network, Dissertation (PhD.), Dept. of Industrial and Systems Engineering, North Carolina State University.
4. Larson R. C. and Odoni A. R., 1981, Urban Operations Research, Prentice-Hall.
5. Daganzo C. F., (1984a) “The Length of Tours in Zones of Different Shapes”, Trans. Res. B 18B, 135-146.
6. Daganzo C. F., 1984b “The Distance Travelled to Visit N Points with a Maximum of C Stops per Vehicle: An Analytical Model and an Application”, Trans. Sci. 18(4), 331-350.
7. Daganzo C. F., 2005, Logistics Systems Analysis, Springer.
8. 2010 census geographic terms and concepts,
http://www.census.gov/geo/www/2010census/GTC_10.pdf
9. Cooper L., 1964, “Heuristic Methods for Location-Allocation Problems”, Society for Industrial and Applied Mathematics, Vol. 6, No. 1, pp. 37-53.

10. Bucci, M. J., 2009, "A Comparison of Meta-Heuristics for Large Scale Facility Location Problems with Economics of Scale", Dissertation (PhD.), Dept. of Industrial and Systems Engineering, North Carolina State University.
11. Kay, M., Matlab Toolbox Matlog. <http://www.ise.ncsu.edu/kay/matlog/index.htm>
12. G. Gutin, A. Yeo and A. Zverovich, Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. Discrete Applied Mathematics 117 (2002), 81-86.
13. 2010 Census Data Products: United States
Web link : <http://www.census.gov/population/www/cen2010/glance/>
14. National Household Travel Survey 2009, Federal Highway Administration, U.S. Department of Transportation. Web link: <http://nhts.ornl.gov/index.shtml>
15. Food Marketing Institute. Web link: <http://www.fmi.org/>
16. Research and Innovative Technology Administration, Bureau of Transportation Statistics. Web link:
http://www.bts.gov/publications/commodity_flow_survey/final_tables_december_2009/index.html
17. UPS Pressroom. Web link: <http://www.pressroom.ups.com/>

18. Ahuja, R. K., Magnanti, T. L. and Orlin, J. B., 1993, Network Flows: Theory, Algorithm and Applications, Prentice Hall.
19. Delaunay Triangulation, Wikipedia. Web link:
http://en.wikipedia.org/wiki/Delaunay_triangulation
20. Hodgson T., 2009, Workbook for Production & Inventory Control, Forecasting & Scheduling.

APPENDICES

Appendix A

MATLAB codes.

1. Script for generating Underlying Road Network.

```
                                %% Generating URN
Triangle=RTAinfo2; % loading block information in structure Triangle.
[XY2,IJD,isXY,isIJD] = subgraph(usrdnode('XY'),...
    isinrect(usrdnode('XY'),boundrect(Triangle.XY,0.1)),...
    usrdlink('IJD'));
                                %%

s = usrdlink(isIJD);
isI = s.Type == 'I'; % Interstate highways
isIR = isI & s.Urban == ' '; % Rural Interstate highways
isIU = isI & ~isIR; % Urban Interstate highways
isR = s.Urban == ' ' & ~isI; % Rural non-Interstate roads
isU = ~isI & ~isR; % Urban non-Interstate roads
                                %%Plot roads

makemap(XY2,0.03) % 3% expansion
h = [];
h = [h pplot(IJD(isR,:),XY2,'r-', 'DisplayName','Rural Roads')];
h = [h pplot(IJD(isU,:),XY2,'b-', 'LineWidth',1.5, 'DisplayName','Urban Roads')];
h = [h pplot(IJD(isI,:),XY2,'k-', 'LineWidth',1.5, 'DisplayName','Interstate Roads')];
                                %%Creating IJT

IJT=IJD;
IJT(isIR,3)=IJD(isIR,3)/70;
IJT(isIU,3)=IJD(isIU,3)/55;
IJT(isR,3)=IJD(isR,3)/45;
IJT(isU,3)=IJD(isU,3)/20;
                                %% Thinning of two degree nodes

pplot(Triangle.XY,'r.')
Triangle.city;
pplot(XY2,'k.')
[tIJD idxIJD]=thin(IJD); [tIJT idxIJT]=thin(IJT);
[tXY,stIJD] = subgraph(XY2,[],tIJD); [tXYT,stIJT] = subgraph(XY2,[],tIJT);
makemap(tXY)
pplot(stIJD,tXY,'c-', 'LineWidth',1)
pplot(tXY,'b.')
pplot(Triangle.XY,'r.')

                                %% Add connector roads from cities to road network
[IJD11,IJD12,IJD22] = addconnector(Triangle.XY,tXY,stIJD);
```

```

h = [h pplot(IJD12,[Triangle.XY; tXY], 'r-', 'DisplayName', 'Connector Roads')];
h = [h pplot(Triangle.XY, 'r.', 'DisplayName', 'Cities')];
IJT22=IJD22; IJT22(:,3)=stIJT(:,3);
IJT12 = IJD12;
IJT12(:,3) = IJD12(:,3)/35; % (IJD11 arcs ignored)
                                %% Shortest paths

n=size([Triangle.XY;tXY],1);
[T,P] = dijk(list2adj([IJT12; IJT22]),1:n,1:n);
                                %% Distance of shortest time route

n=size(Triangle.XY,1);
A = list2adj([IJD12; IJD22]);
D = zeros(n);
for i = 1:n
    i
    for j = 1:n
        D(i,j) = locTC(pred2path(P,i,j),A);
    end
end
                                %% Find shortest path

TotalXY=[Triangle.XY;tXY];
idx1 = 14; idx2 = 186;
[t,p] = dijk(list2adj([IJT12; IJT22]),idx1,idx2);
h(6)=pplot({p},TotalXY,'r-', 'LineWidth',4, 'DisplayName', 'Shortest Path');
pplot(TotalXY([idx1 idx2],:),Triangle.city([idx1 idx2]))
title(sprintf('
    From %s to %s: Distance = %.2f mi, Travel Time = %.1f hrs',...
    Triangle.city{idx1 },Triangle.city{idx2 },D(idx1,idx2),t));

```

2. [*allocation.m*] MATLAB function for finding allocations

```
function [alloc DC]=allocation(DC,D)
% finds allocation of EFs to NFs
% DC = input of 1-by-m array of indices of NFs
% D = n by n array of time required to travel shortest path distances
% alloc = n-by-n logical matrix of allocations
% DC = returns indices of only those NFs which are allocated to at least
% one EF

n=size(D,1); m=size(DC,2); alloc=zeros(n,size(D,2));
i=1;
while(i<=n)
[Y I]=min(D(i,DC));
I=I(1,1); idx=DC(1,I);
alloc(i,idx)=1; i=i+1;
end
S=sum(alloc,1); S=S(1,DC); DC(S==0)=[];
```

3. [*LoadUnload.m*] MATLAB function for calculating loading-unloading time.

```
function LUtime=LoadUnload(LU,sP)
% finds total loading-unloading time of package transported from DC i to DC
% j
% LUtime = n by n array of loading-unloading times
% LU = scalar value of loading-unloading time required for one truck/trip
% sP = n by n array of shortest paths

i=1;
while(i<=size(sP,1))
    j=1;
    while(j<=size(sP,2))
        if(i~=j)
            path=pred2path(sP,i,j);
            LUtime(i,j)=(LU*2*(size(path,2)-1))/60;
        end
        j=j+1;
    end
    i=i+1;
end
```

4. [*connector.m*] MATLAB code for adding connectors to network of DCs

```
function xIJD=connector(T,b,d0,thresh,c)
% Adds new arcs to the network formed after Delaunay Triangulation
% xIJD = n by 3 array containing n arcs (i,j) and lengths of arcs
% T = Delaunay Triangulation of original network
% b = actual indices of selected nodes from the original network
% d0 = shortest distance matrix between all the nodes in the network (DC's)
% thresh = Parameter (threshold to add an arc in the network) [thresh=0.85]
% c= the shortest distance matrix between all the nodes of reduced new network obtained by
% Delaunay Triangulation
% This function was created by Amogh Bansal. For more information please
% refer [2].

N=trineighbors(1:size(T,1),T);
xIJD=[];
for i=1:size(N,1)
    for j=1:size(N,2)
        if(N(i,j)~=NaN & N(i,j)>i)
            x=setxor(T(i,:),T(N(i,j),:));
            do=d0(b(x(1)),b(x(2)));
            dt=c(x(1),x(2));
            if(do<thresh*dt)
                xIJD=[xIJD;x(1),x(2),do];
            end
        end
    end
end
end
```

5. [*DemOnArc.m*] MATLAB function for calculating demand on each arc in network of DCs

```

function [DemArc DemArc1]=DemOnArc(P,demand)
% finds demand flow on each arc in the network of DCs
% DemArc = n by n array of demand of packages on arcs ij
% DemArc1 = m by 3 array containing arcs ij and demands d
% P = n by n array of shortest paths
% demand = n by n array of demand flowing from DC i to DC j
DemArc=zeros(size(demand,1));
i=1;
while(i<=size(demand,1))
    j=1;
    while(j<=size(demand,2))
        if(i~=j)
            path=pred2path(P,i,j); n=size(path,2);
            k=1;
            while(k<=(n-1))
                DemArc(path(1,k),path(1,k+1))=demand(i,j)+DemArc(path(1,k),path(1,k+1));
                k=k+1;
            end
        end
        j=j+1;
    end
    i=i+1;
end

i=1; z=1;
while(i<=size(DemArc,1))
    j=1;
    while(j<=size(DemArc,2))
        if(DemArc(i,j)~=0)
            DemArc1(z,1)=i; DemArc1(z,2)=j; DemArc1(z,3)=DemArc(i,j);
            z=z+1;
        end
        j=j+1;
    end
    i=i+1;
end
end

```

6. [*Proxfac.m*] MATLAB function for finding proximity factors.

```
function W = Proxfac(D,w,p)
%PROXFAC Order-based proximity factor.
% W = proxfac(D,w,p)
%   D = n x n distance matrix
%   w = n-element marginal weight
%   p = proximity factor (p == 0 => no proximity adjustment)
%       = 2.572351526092777, default (factor for 752mi LTL avg dist)
%   W = n x n weight matrix, where, for p = 0, W(i,j) = w(i)*w(j)
%
% Default based on LS fit to 1997 State-to-State Commodity Flow data
% on the value of shipments (http://www.bts.gov/ntda/cfs/cfs97od.html)
% with 95% confidence interval [6.4139, 6.6986].

% Copyright (c) 1994-2003 by Michael G. Kay
% Matlog Version 7 25-Sep-2003

% Input Error Checking *****
if nargin < 3 | isempty(p), p = 2.572351526092777; end
% End (Input Error Checking) *****

m = length(D);
W0 = w(:)*w(:)';
I = argsort(argsort(D,2),2);
I = triu(I) + triu(I,1)';
R = ((1-p/m).^(I-1))/(mean((1-p/m).^((1:m)-1)));
W = R.*W0./sum(sum(R.*W0));
```

7. [*remArk.m*] MATLAB function for removing arcs from network of DCs.

```
function remidx=remArc(IJD)
% finds indices of the arcs to be removed from network
% remidx = indices of arcs to be removed from network
% IJD = n by 3 array containing arcs ij and length of those arcs
i=1; remidx=[0];
while(i<=size(IJD,1))
    idx=IJD(i:i+2,3)==max(IJD(i:i+2,3));
    if(IJD(idx,3)>(0.9*sum(IJD(~idx,3),1)))
        temp=[i:i+2]*idx;
        temp(temp==0)=[];
        remidx=[remidx; temp];
    end
    i=i+3;
end
if(size(remidx,1)==1)
    remidx=[];
end
if(size(remidx,1)>1)
    remidx=remidx(2:end,:);
end
```

8. [*sqarea.m*] MATLAB function for finding area of square enclosing blocks allocated to a DC.

```
function R=sqarea(XY)
% XY = co-ordinates of blocks
% R = area of square enclosing XY.
Xmax=max(abs(XY(:,1))); Xmin=min(abs(XY(:,1)));
Ymax=max(abs(XY(:,2))); Ymin=min(abs(XY(:,2)));
R=(dists([Xmax,0],[Xmin,0],'mi')*dists([0,Ymax],[0,Ymin],'mi'));
```

9. [*tri2IJ.m*] MATLAB code for converting Delaunay Triangulations to IJ form.

```
function IJ=tri2IJ(tri)
% transforms output of Delaunay triangulation to IJ matrix
% tri = n by 3 array containing vertices of n triangles
% IJ = m by 2 array containing arcs ij in network of Delaunay Triangulation

i=1; z=1;
while(i<=size(tri,1))
IJ(z,1)=tri(i,1); IJ(z,2)=tri(i,2); z=z+1;
IJ(z,1)=tri(i,2); IJ(z,2)=tri(i,3); z=z+1;
IJ(z,1)=tri(i,1); IJ(z,2)=tri(i,3); z=z+1;
i=i+1;
end
n=size(IJ,1);
IJ(n+1:2*n,1)=IJ(1:n,2); IJ(n+1:2*n,2)=IJ(1:n,1);
```

10. [*WFTT.m*] MATLAB code for calculating waiting time for truck.

```
function WT=WFTT(sP,headway)
% WT = n by n array of waiting time where n is number of points in network
% sP = n by n array giving shortest path from point i to point j
% headway = n by n array of headways on arc ij.
WT=zeros(size(sP,1));
i=1;
while(i<=size(sP,1))
    j=1;
    while(j<=size(sP,2))
        if(i~=j)
            path=pred2path(sP,i,j); n=size(path,2); k=1;
            while(k<=(n-1))
                WT(i,j)=WT(i,j)+headway(path(1,k),path(1,k+1));
                k=k+1;
            end
        end
        j=j+1;
    end
    i=i+1;
end
```

11. [*BestSol.m*] MATLAB code for finding starting DCs from *IniGuess.mat* which will be an input to heuristic.

```
function [TC2,DC2]=BestSol(i,pcd,LU,HWratio)
% randomly selects i points from IniGuess data file as DC locations and calculates
% network cost for those locations. Repeats the procedure for 100 times and
% returns set of DCs which gives minimum cost.
% i = number of DCs to be selected
% pcd = per capita demand
% LU = assumed value for loading-unloading time (0,5,10)
% TC2 = minimum cost found after 100 trials
% DC2 = co-ordinates of i DCs corresponding to lowest cost

startup(1);
load('blocksDTP'); load('blocksPLN'); load('RTAblockinfo'); load('IniGuess');

opt.DC=zeros(100); opt.TC=zeros(100,1); opt.TC(:,1)=inf;
z=1;
while(z<=100)
    A=randperm(201);
    DC=blocksPLN.XY(IniGuess(A(1,1:i),:),:);
    [TC,~,~,results]=contNetCost2(DC,pcd,LU,HWratio);
    if (TC<0)
        TC=inf; end
    if (sum(results<0,2)>0)
        TC=inf; end
    if(TC<=opt.TC(i,1))
        opt.TC(i,1)=TC; opt.DC(i,1:i)=DC(:,1); opt.DC(i,i+1:(2*i))=DC(:,2);
    end
    z=z+1;
end

TC2=opt.TC(i,1); DC2=opt.DC(i,1:2*i);
```

12. [*Preprocessing.m*] MATLAB code for *preprocessing*.

```

load('PLN'); load('DTP'); load('RTAblockinfo'); load('blockarea');

idx=(((s.pop*pcd)/14)<180).*[1:925]'; idx(idx==0)=[];
sx=PLN.XY(idx,1); sy=PLN.XY(idx,2); stri=delaunay(sx,sy);
i=1;
while(i<=size(stri,1))
    j=1;
    while(j<=size(stri,2))
        stri(i,j)=idx(stri(i,j),1); j=j+1;
    end
    i=i+1;
end
i=1; freq=[];
while(i<=size(idx,1))
    freq(i,1)=sum(sum(stri==idx(i,1),1),2); i=i+1;
end
grp=1; finclus=[];
while(sum(sum(stri,1),2)>0)
    pt1=(freq==min(freq)).*[1:size(freq,1)]'; pt1(pt1==0)=[]; pt1=pt1(1,1);
    i=1; nbor=[];
    while(i<=3)
        nbor=[nbor;stri(:,i).*sum(stri==idx(pt1(1,1),1),2)]; i=i+1;
    end
    nbor(nbor==0)=[]; nbor=unique(nbor);
    nbor(:,2)=DTP.T(idx(pt1(1,1)),nbor(:,1));
    perhrdem=((s.pop*pcd)/14);
    nbor(:,3)=perhrdem(nbor(:,1));

    if(sum(nbor(:,3),1)>=180)
        add=0; i=1; cluster=[];
        while(add<180)
            cluster(i,1)=grp;
            cluster(i,2)=nbor((nbor(:,2))==min(nbor(:,2))),1);
            add=add+nbor((nbor(:,2))==min(nbor(:,2))),3);
            cluster(i,3)=nbor((nbor(:,2))==min(nbor(:,2))),3);
            nbor((nbor(:,2))==min(nbor(:,2))),:)=[]; i=i+1;
        end
        finclus=[finclus; cluster];
        i=1;
        while(i<=size(cluster,1))
            stri(stri==cluster(i,2))=0;

```

```

    freq((idx==cluster(i,2)),1)=Inf;
    i=i+1;
end

grp=grp+1;
else
    i=1;
    while(i<=size(nbor,1))
        grpno=finclus((DTP.T(nbor(i,1),finclus(:,2))'==min(DTP.T(nbor(i,1),finclus(:,2))')),1);
        finclus=[finclus;[grpno nbor(i,1) nbor(i,3)]];
        freq((idx==nbor(i,1)),1)=Inf;
        stri(stri==nbor(i,1))=0;
        i=i+1;
    end
end
end

i=1;
while(i<=max(finclus(:,1)))
    cluster=finclus((finclus(:,1)==i),:);
    clusblk(i,1)=sum((PLN.XY(cluster(:,2),1).*cluster(:,3)),1)/(sum(cluster(:,3),1));
    clusblk(i,2)=sum((PLN.XY(cluster(:,2),2).*cluster(:,3)),1)/(sum(cluster(:,3),1));
    clusidx(i,1)=i;
    cluspop(i,1)=sum(s.pop(cluster(:,2),1),1);
    clusarea(i,1)=sum(blockarea(cluster(:,2),1),1);
    i=i+1;
end

```

13. [contNetCost2.m] MATLAB function to calculate average delivery time.

```
function [TC W results IJ ]=contNetCost2(DC,pcd,LU,HWratio)

% finds cost of the network for given DC locations
% TC = average time needed to deliver a package
% W = n by 925 array of allocations of 925 blocks to n DCs
% results = 1 by 6 array containing mean values of waiting time, loading-unloading time,
%     DC 2 DC travel time, local travel time, local truck waiting time and total delivery time
%     respectively in each column
% IJ = IJDT matrix of network of DCs'
% DC = co-ordinates of points where DCs will be located
% pcd = per capita demand
% LU = assumed value for loading-unloading time (0,5,10)

startup(1);
load('PLN'); load('DTP'); load('RTAblockinfo'); load('newDC');
[alloc DC]=allocation(DC,blocksDTP.T(1:925,:));

% Delaunay Triangulation and conversion to IJDT
x=blocksPLN.XY(DC,1); y=blocksPLN.XY(DC,2); x=x'; y=y';
tri=delaunay(x,y);
IJ=[];
IJ=tri2IJ(tri);
i=1;
while(i<=size(IJ,1))
IJ(i,3)=blocksDTP.D(DC(1,IJ(i,1)),DC(1,IJ(i,2)));
IJ(i,4)=blocksDTP.T(DC(1,IJ(i,1)),DC(1,IJ(i,2))); i=i+1;
end
%%%%
%% shortest time paths and corresponding distances
n=size([x;y]',1); sT=[]; sP=[];
[sT,sP] = dijk(list2adj([IJ(:,1:2) IJ(:,4)]),1:n,1:n);

n=size([x;y]',1);
A = list2adj(IJ(:,1:3));
sD = zeros(n);
for i = 1:n
    for j = 1:n
        sD(i,j) = locTC(pred2path(sP,i,j),A);
    end
end
end
```


14. [NSALA.m] MATLAB code for heuristic.

```
function [optTC,optDC]=NSALA(i,pcd,LU,HWratio)
% finds near-optimal solution minimizing mean delivery time when i number
% of DCs located
% i = number of DCs to be located
% pcd = per capita demand
% LU = assumed value for loading-unloading time (0,5,10)
% optTC = minimum value of delivery time
% optDC = indices of DCs which minimize the cost

load('blocksPLN'); load('opt-348-5-17nu');
DC(1:i,1)=opt.DC(i,1:i); DC(1:i,2)=opt.DC(i,i+1:2*i);
[TC W DC1]=contNetCost2(DC,pcd,LU,HWratio); optTC=TC; optDC=DC1.dc';
done=0;
while(~done)
row=1;
while(row<=size(DC1.dc,1))
idx=W(row,:).*[1:925]; idx(idx==0)=[];
L=1;
while(L<=size(idx,2))
tempDC=optDC; tempDC(1,row)=idx(1,L);
[tempTC,~,~,results]=contNetCost2(blocksPLN.XY(tempDC,:),pcd,LU,HWratio);
if (tempTC<0)
tempTC=inf; end
if (sum(results<0,2)>0)
tempTC=inf; end
optTC=min(tempTC,optTC);
if (optTC==tempTC)
optDC=tempDC;
end
L=L+1;
end
row=row+1;
end
if((TC-optTC)>=0.5)
[TC W DC1]=contNetCost2(blocksPLN.XY(optDC,:),pcd,LU,HWratio); done=0;
optTC=TC; optDC=DC1.dc';
else
done=1;
end
end

end
```

15. [VRforClusters.m] MATLAB function for route design for groups

```
function [routedist, WT] = VRforClusters(pcd,W,dc,DC)

load('PLN'); load('DTP'); load('RTAblockinfo'); load('blockarea');
if(pcd==2.29)
load('clusarea229'); load('cluspop229'); load('clusidx229'); load('clusblk229');
end
if(pcd==3.48)
load('clusarea348'); load('cluspop348'); load('clusidx348'); load('clusblk348');
end
if(pcd==5.28)
load('clusarea528'); load('cluspop528'); load('clusidx528'); load('clusblk528');
end
WT=0; insideblk=0;
idx180=(W(dc,:).*[1:81])'; idx180(idx180==0)=[];
if((size(idx180,1)+size(idx180,2))>1)
demand=(cluspop(idx180,:)*pcd)/14;
r=(dists(clusblk(idx180,:),PLN.XY(DC,),'mi').*(cluspop(idx180,1)*pcd))./sum((s.pop*pcd),
1);
r=sum(r,1);
perhrdem=(cluspop(idx180,1)*pcd)/14;
i=1;
while(i<=size(idx180,1))
    if(perhrdem(i,1)<=240)
        time=(1.7*(perhrdem(i,1)*14))/sum((s.pop*pcd),1);
        WT=WT+time;
        inb=(0.57*1.5*sqrt(clusarea(idx180(i,1),1)*perhrdem(i,1)))/20;
        inb=(inb*(perhrdem(i,1)*14))/(2*sum((s.pop*pcd),1));
        insideblk=insideblk+inb;
    end
    if(perhrdem(i,1)>240)
        time=(14/((perhrdem(i,1)*14)/240))*1.7;
        time=(time*(perhrdem(i,1)*14))/sum((s.pop*pcd),1);
        WT=WT+time;
        inb=(0.57*1.5*sqrt(clusarea(idx180(i,1),1)*240))/20;
        inb=(inb*(perhrdem(i,1)*14))/(2*sum((s.pop*pcd),1));
        insideblk=insideblk+inb;
    end
    i=i+1;
end
routedist=r+insideblk;
else routedist=0; WT=0; end
```

16. [VRforNonClusters.m] MATLAB function for route design for blocks

```
function [routedist, WT] = VRforNonClusters(pcd,W,dc,DC)

load('PLN'); load('DTP'); load('RTAblockinfo'); load('blockarea');

WT=0; insideblk=0;
idx=(W(dc,:).*[1:925]'); idx(idx==0)=[];
demand=(s.pop(idx,)*pcd)/14;
idx180=idx.*(demand>=180); idx180(idx180==0)=[];
r=(DTP.T(idx180,DC).*(s.pop(idx180,1)*pcd))./sum((s.pop*pcd),1);
r=sum(r,1);

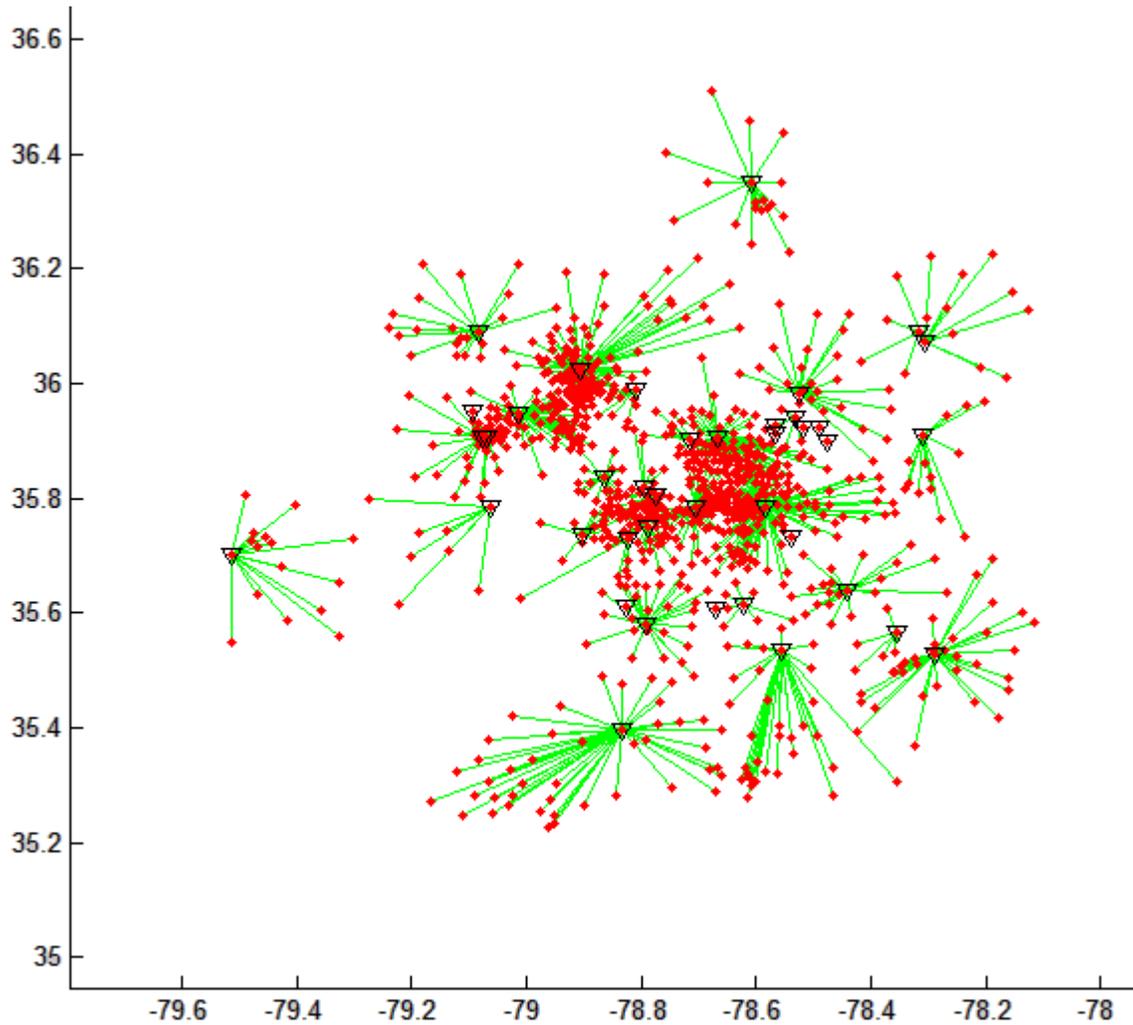
perhrdem=(s.pop(idx180,1)*pcd)/14;
i=1;
while(i<=size(idx180,1))
    if(perhrdem(i,1)<=240)
        time=(1.7*(perhrdem(i,1)*14))/sum((s.pop*pcd),1);
        WT=WT+time;
        inb=(0.57*1.5*sqrt(blockarea(idx180(i,1),1)*perhrdem(i,1)))/20;
        inb=(inb*(perhrdem(i,1)*14))/(2*sum((s.pop*pcd),1));
        insideblk=insideblk+inb;
    end
    if(perhrdem(i,1)>240)
        time=(14/((perhrdem(i,1)*14)/240))*1.7;
        time=(time*(perhrdem(i,1)*14))/sum((s.pop*pcd),1);
        WT=WT+time;
        inb=(0.57*1.5*sqrt(blockarea(idx180(i,1),1)*240))/20;
        inb=(inb*(perhrdem(i,1)*14))/(2*sum((s.pop*pcd),1));
        insideblk=insideblk+inb;
    end
    i=i+1;
end

routedist=r+insideblk;
```

Appendix B

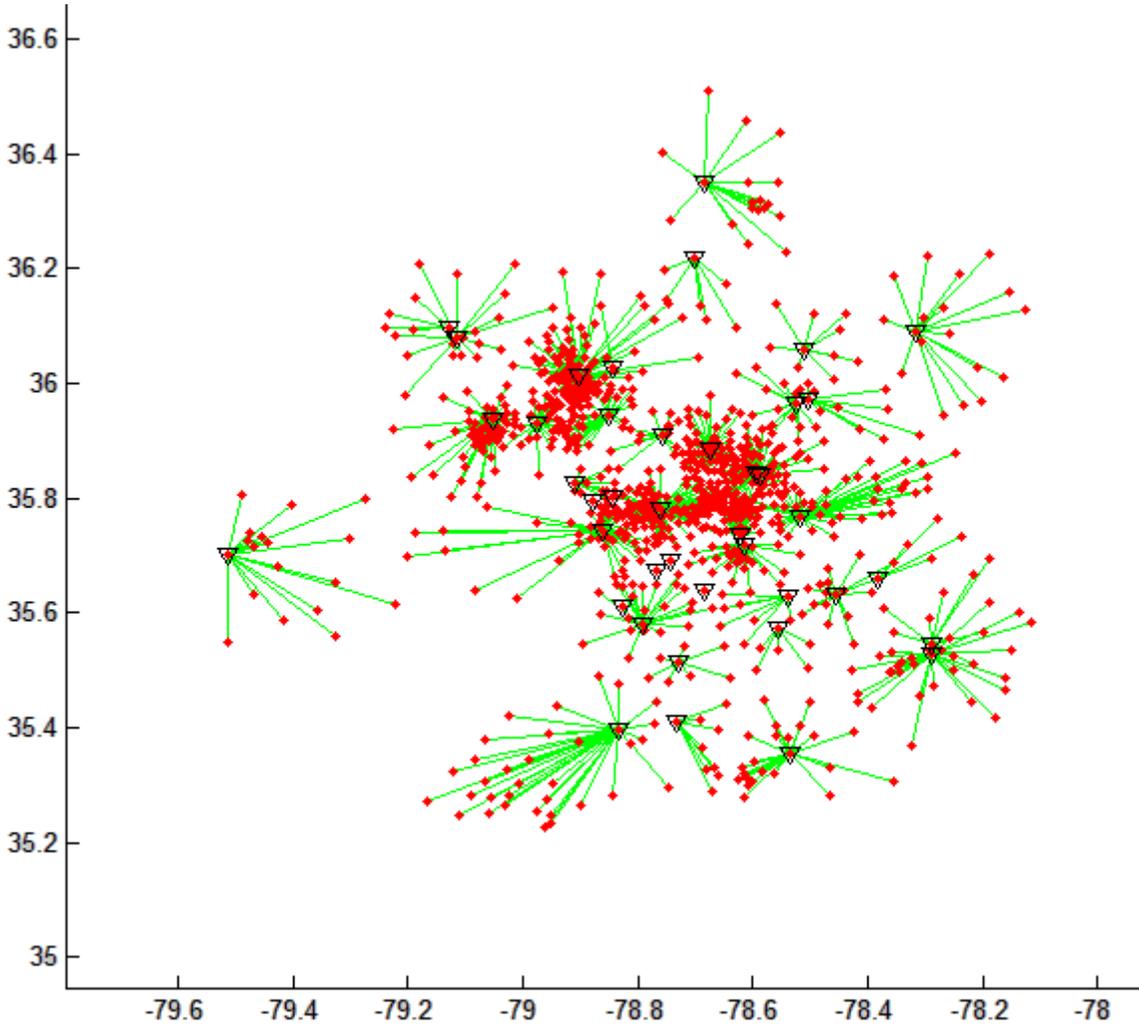
Optimal DC location and allocation plots for different values of pcd and LU .

1. $Pcd = 5.28$ and $LU = 10$ minutes



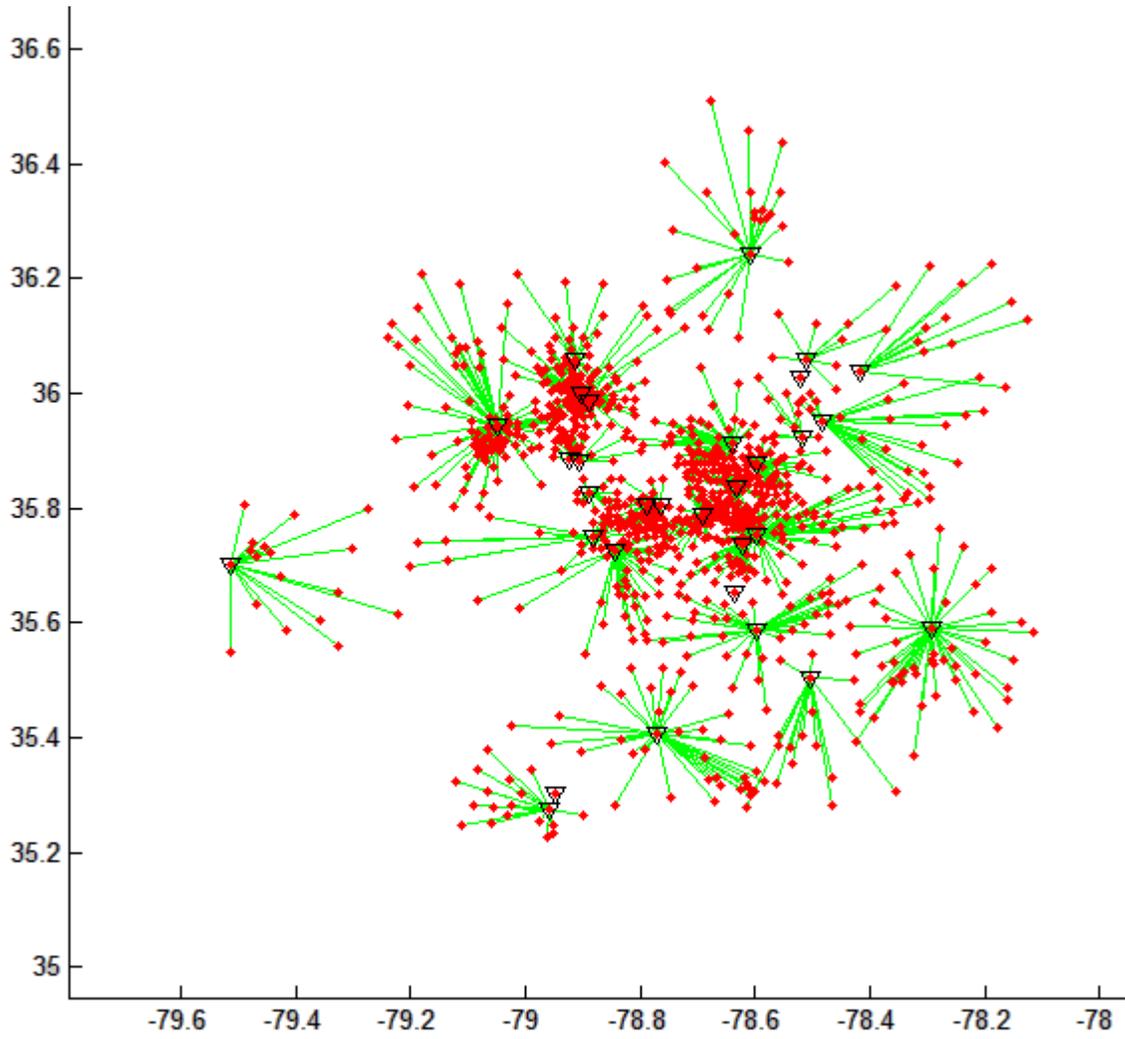
Average Delivery Time = 128 minutes
Number of DCs = 40

2. $Pcd = 5.28$ and $LU = 5$ minutes



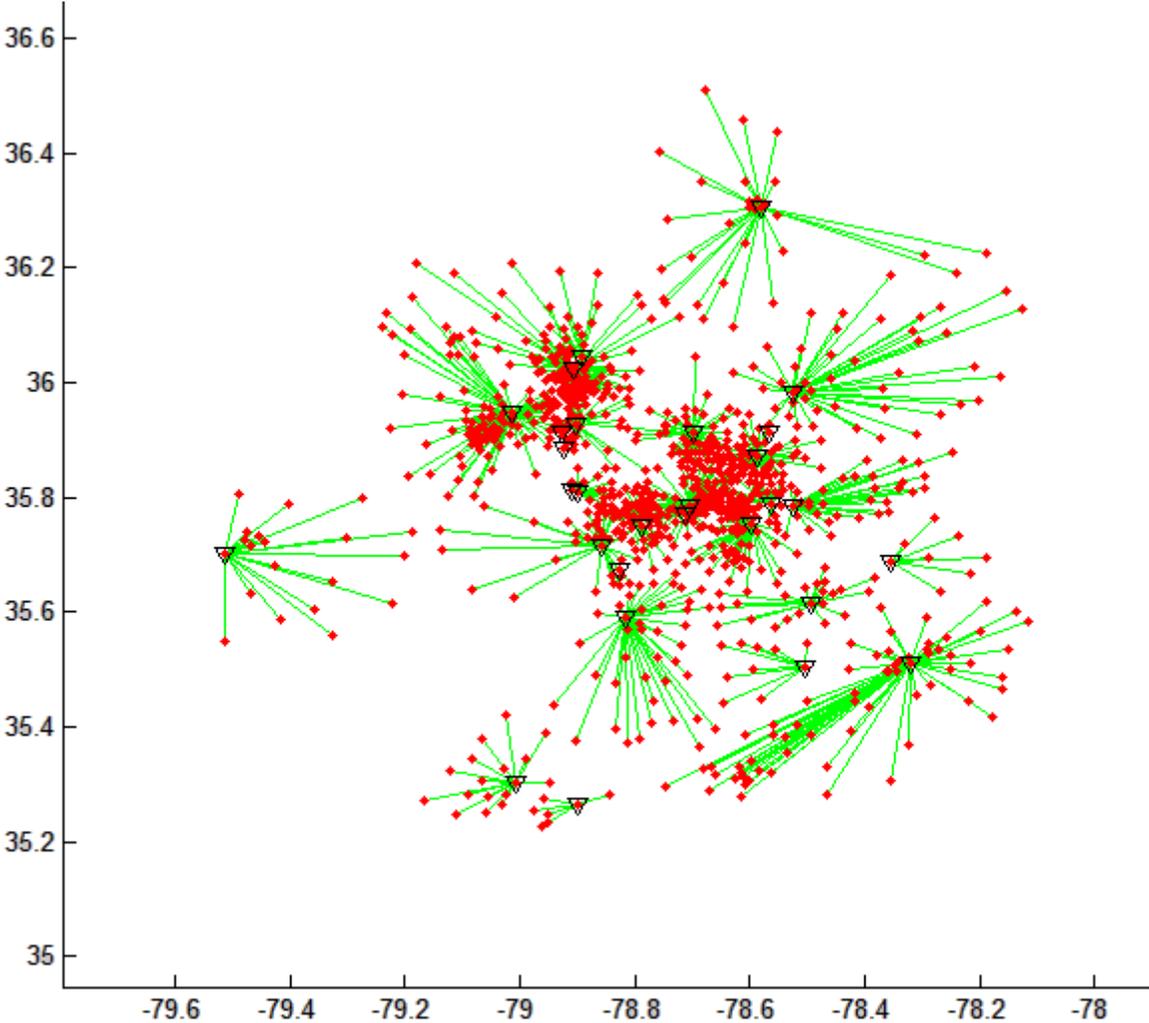
Average Delivery Time = 121 minutes
Number of DCs = 41

3. $Pcd = 2.29$ and $LU = 0$ minutes



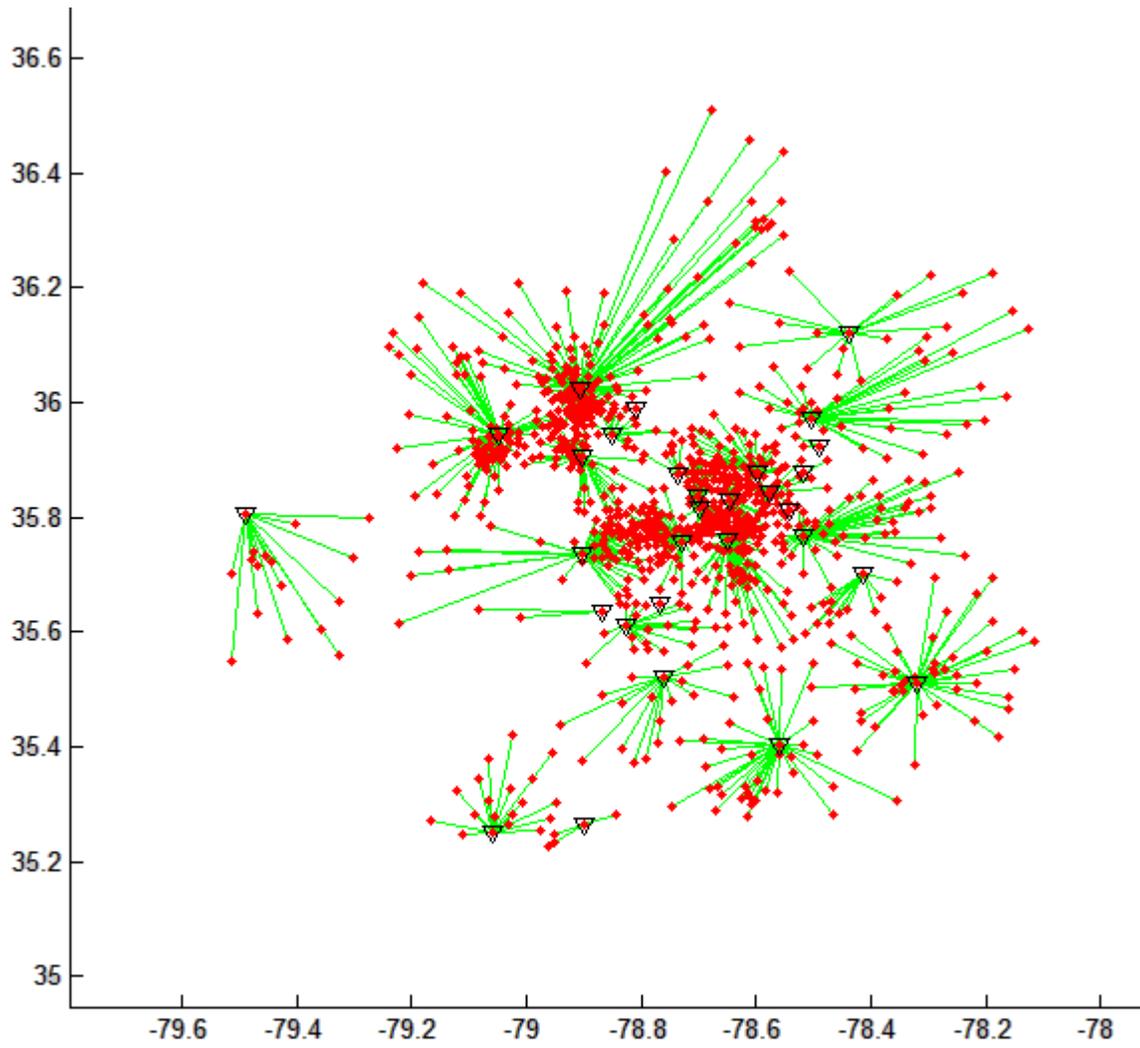
Average Delivery Time = 155 minutes
Number of DCs = 32

4. $Pcd = 2.29$ and $LU = 5$ minutes



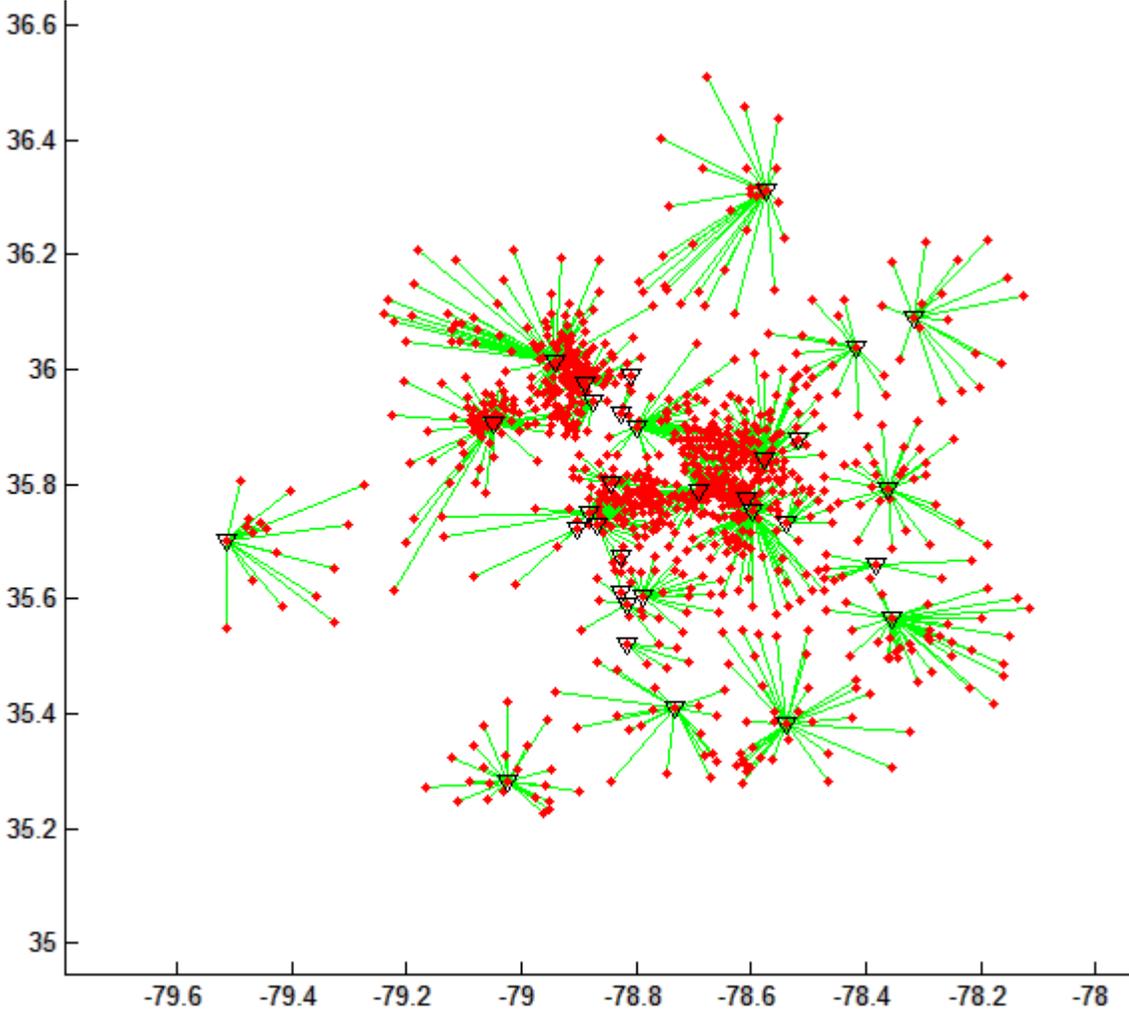
Average Delivery Time = 159 minutes
Number of DCs = 30

5. $Pcd = 3.48$ and $LU = 10$ minutes



Average Delivery Time = 147 minutes
Number of DCs = 31

6. $Pcd = 3.48$ and $LU = 5$ minutes



Average Delivery Time = 140 minutes
Number of DCs = 32

7. Groups/Clusters for $pcd = 5.28$

