Location Allocation Report

Locating prospective Lowe's stores in Waterloo, Ontario

Eric Josef	Schmitz
Jiachen	Wei
Timothy Vi Triet	Luong

Wednesday March 22, 2017

University of Waterloo, ON, Canada

GEOG 381

Winter 2017

Assign-ment 4

Table of Contents

Introduction

I. Problem Statement

Lowe’s home improvement stores have come a long way from a single store in North Carolina. Today it has become one of the largest and most dominant hardware and home renovation stores in the United States. Lowe’s is currently looking to expand into the Canadian market. Their operation currently involves 34 stores in Canada, but the corporation wants to expand to around 100 stores. Lowe’s needs to decide where the best locations for these new stores are. It is our team’s job to assess the demand for newly proposed Lowe’s store locations in the Waterloo region, Ontario Canada. Including a store that will be located on Ira Needles Boulevard in Waterloo. It is an important business decision by Lowe's to assess the demand of proposed store locations. By doing this, Lowe’s will be able to maximize profits on a limited number of proposed stores, because Arc GIS tools can assess how many people will be able to make the trip to that store. Without this type of data, Lowe’s could choose a location for a store that would not perform as well as another proposed location. To find our answer we will look at three different scenarios. The first scenario is the base scenario where the only demand evaluated, is at existing competitor stores. The second scenario evaluates all potential Lowe’s store locations to choose a single optimal site. The third scenario evaluates all potential Lowe’s store locations like the last example; the difference is this time the goal is to choose two sites. Our assessment was completed with network analysis, spatial interaction models, and location-allocation modeling. A unique part of this report is the inclusion of python coding over all aspects of the assessment. The tools and their role in the assessment is discussed below.

II. Tools we Use

Our assessment of demand is done through ArcGIS. Network analysis in ArcGIS gives us the ability to determine the least-cost network paths between different points and locations. In our report, we use travel time as the factor our network analysis is based off. Network analyst can also be used to determine things such as shortest paths, travel time, travel cost, and distance to different types of key locations. This makes network analysis very important when it comes to determining retail locations since the path between the store and consumer is a very important factor to how likely a consumer is to come to the store. We set up our network analyst so that the store location with the highest amount of demand from areas of the Waterloo Region within a 17-minute drive would be the optimal choice. This decision made store locations that are in relation to high clusters of residence and fast transportation routes more valuable. Location is one of the most important factors in this report. The location quality of each site is what determines the demand of the store. Location-allocation analysis is used in ArcGIS to match the store locations to demand points on the map. Demand points are manually created points assigned to areas around the Waterloo Region to represent the quality of population in that area. We manually create these points with an equation based off of income split into quintiles. This means that each demand point represents a different population of income status in our example. It is our belief that those with a higher income are more likely to go to a home improvement store, so those demand points are higher quality. The next step after creating these demand points is for the program to allocate each demand point to a store location that is most compatible with it based on travel time. So the more demand points that are allocated to a store location, the higher demand for the store. After it is all connected, it creates a spatial interaction model that represents the connection between demand and store locations. Python has been incorporated into this analysis. Python is language of code; it interacts with GIS by allowing us to write a line of code to create our demand points, network datasets, and rest of the assessment instead of using Arc GIS's tools. By using Python, we are able to recreate the assessment by running the code, and anyone else can theoretically recreate it by running the code as well. This is an important factor because it simplifies many things for future projects with Lowe's. This spatial interaction data is a necessity for evaluating demand. Although, Arc GIS cannot evaluate some of the more topographical data, for that external information is required. Background info of the store locations is discussed below.

III. Background Information on Stores

In our analyses, there are five potential Lowe's store locations, including the one to be located at Ira Needles Boulevard. For an accurate representation of demand at these Lowe's stores versus other home hardware stores in the area, we include twelve large competitor's stores. It is in the interest of Lowe's for this assessment to give as accurate of a representation of demand as possible. To do this, we have to look at certain information that the network data set cannot take into account such as topographical information. All of our potential sites are located on the outer limits of highly clustered residential areas of either Waterloo, Kitchener, or Cambridge. They are all on or very close to a major transportation route, which is great for the increased amount of impulse buyer traffic that can go by. This transportation network can also cut travel time with higher speed limits and reduce the amount of time spent at red lights and stop signs. The problem with this though can be the risk of traffic jams during rush hours. There many types of factors that affect the demand of a store location. Individual shoppers are heavily persuaded by many different things, which will be discussed further in the report below. It is valuable to Lowe's to have the most accurate store demand data possible. To get this kind of data we have looked at different types of variables, factors, future outcomes, and how they interact with the store locations.

Methods

I. Creating Network Dataset

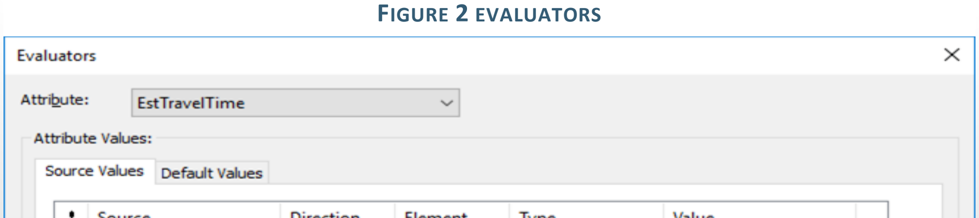
The network dataset is going to be created from the RMoWFD (Regional Municipality of Waterloo Feature Dataset) feature class. The travel time has already been calculated in this data set as opposed to requiring calculation in the Traveltm column found in assignment 3's RMoWFD, therefore, we can go to ArcCatalog and create a new network dataset on RMoWFD right away. The streets feature class is the one that will be participating in the creation of the network dataset, with default values being used until the attribute specification window. In this window, two additional attributes will be created: EstTravelTime, a cost attribute based on Minutes, while OneWay is a restriction attribute, illustrated by Figure 1 attributes.

FIGURE 1 ATTRIBUTES

Name	Usage	Units	Data Type
EstTravelTime	Cost	Minutes	Double
Length	Cost	Meters	Double
OneWay	Restriction	Unknown	Boolean

Evaluators for EstTravelTime must be created of type 'Field', with the value being 'Traveltm', illustrated by Figure 2.

FIGURE 2 EVALUATORS

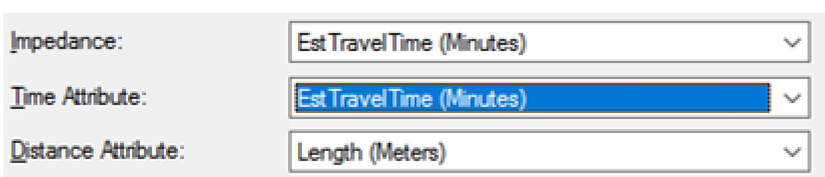


The evaluators for OneWay are of type script, with the value of the script representing whether or not it is a two-way (0) or one-way street, where if it is a one-way street, it is a value of 1 or 2, where 1 represents the direction of the road being the direction that the feature is digitized (From-To), while 2 is the opposite direction (To-From). The script used for this value calculation is:

```
result=false
if [ONEWAY]=X then
        result=true
end if
```

where X is the value 1 or 2, where 1 is for the From-To direction, while 2 is for the To-From direction. In the next page, we will set the values like Figure 3.

FIGURE 3 SETTING IMPEDANCE



In the next page, a Name field will be specified for the edge source of the streets feature class. In the directions window, FULL_NAME will be set as the name for the 'Primary' rank of streets like in Figure 4, and then the network dataset was created.

FIGURE 4 PRIMARY RANK

Rank	Prefix	Prefix Type	Name	Suffix Type	Suffix	Fu
Primary			FULL_NAME			FU

II. Creating Relative Path and Additional Competitor Store List

For sections 2.2 to 2.6, we will be referring to the sections 3.X, where X is of the same numerical value in the "mainScript.py" Python script. A root path variable will be created using os.path.dirname(sys.argv[0]). Os.path.dirname returns the address of which the Python script is being run from, while sys.argv[0] will return this value as a string to be assigned to the variable "root". An environment workspace is then created using the root path and appending "\\RMOW.gdb". This points to the RMOW geodatabase folder that is included in the submission. Arcpy.env.overwriteOutput will allow any arcpy functions to overwrite duplicate entities that exist within the gdb folder.

III. Geocode Additional Competitor Stores

A path variable is created for the additional competitor stores to be added to the Network Analysis: root + "\\a4_stores.csv". Several parameter variables will also be created to be used in arcpy functions: Reference_data, addressLocator, outAddresses, fieldMap, locator_style. Reference_data is an address that points to the attribute table inside RMoWFD's streets feature class. addressLocator is an address that points to the output Address Locator, which allows interpretation of textual addresses into features that can be pointed to on ArcGIS. outAddresses is the variable for the address to the to be created feature class newStores. fieldMap is a string parameter that connects columns in the reference_data streets attribute table to the columns in the locator_style "US Address - Dual Ranges" parameters. CreateAddressLocator_geocoding is an arcpy function that takes the locator_style, and reference_data and connects them together using the fieldMap parameter, outputting the Address Locator to the addressLocator variable. GeocodeAddresses_geocoding is an arcpy function that connects the storesToGeocode csv table to the addressLocator list of addresses, which outputs the newly classed features to the outAddresses feature class.

IV. Merging Existing and New Stores

This section of Python code will combine the outAddresses feature class to the existing addresses in the ExistingBldgCentres feature class, or origStores. Three variables are created: fields, fieldNames, and outStorestmp. fields is a variable which contains a list containing Field objects, or columns in the attribute table of a feature class-in this case, outAddresses, or the new stores. fieldNames is a list of fields that are extracted from fields, which are field objects in a readable string format. outStorestmp is a variable that is a temporary representation of the ExistingBldgCentres feature class. In the for loop, the strings in the list used by the for loop are removed from fieldNames. In arcpy.DeleteField_management, the remaining fieldNames are removed from outAddresses, leaving the original fields that can be found in the for loop. In arcpy.Merge_management, origStores and outAddresses are joined into one table which is output into outStorestmp, while arcpy.Delete_management and arcpy.Rename_management deletes the old origStores feature class and then replaces it with outStorestmp which now becomes origStores.

V. Estimating Demand

In order to perform Network Analysis in ArcGIS, we need to create a feature class that represents each CDA (polygon) as a point with an attribute called demand. In our case, demand is defined as the expenditure (in Canadian dollars) the residents in one CDA spend on home improvement.

The available demographics in each Waterloo CDA are population by sex and age group, income levels by household and age-grouped population by education level, field of study and location of study. We retrieve from Statistics Canada Table 203-0022 (2011-2015) Canadian household total expenditure by income quintiles (each quintile representing 20% of the population), and Canadian household spending on furnishing and equipment by the same income quintiles. This data file is attached as Expenditure_Based_Income.csv. However, we decide not to include other data such as education levels (even though available from Statistics Canada) in our demand calculation because income and education may not be interdependent variables to formulate one equation. As shown in Table 1, we calculate in Excel for each income quintile average household expenditure and average spending on furnishing and equipment from 2011 to 2015.

TABLE 1 AVERAGE HOUSEHOLD AND FURNISHING EXPENDITURE (2011-2015)

Table 203-0022 Survey of household spending (SHS), household spending, Canada, regions and provinces, by household income quintile, annual (dollars)(1,3,4,7,8)									
Survey or program details:									
Survey of Household Spending - 3508									
Geogra	Statistic	Before-tax household ir	Household expenditur	2011	2012	2013	2014	2015	AVERAGE
Canad	Average expenditure per household	Lowest quintile	Total expenditure	29214	30028	31410	31974	33705	31266.2
Canad	Average expenditure per household	Lowest quintile	Household furnishings	833	978	801	1039	855	901.2
Canad	Average expenditure per household	Second quintile	Total expenditure	45235	43850	47965	47295	49750	46819
Canad	Average expenditure per household	Second quintile	Household furnishings	1487	1509	1295	1203	1552	1409.2
Canad	Average expenditure per household	Third quintile	Total expenditure	63032	63756	66546	67829	68893	66011.2
Canad	Average expenditure per household	Third quintile	Household furnishings	1747	1856	2005	1908	2066	1916.4
Canad	Average expenditure per household	Fourth quintile	Total expenditure	86821	88186	92886	94726	96493	91822.4
Canad	Average expenditure per household	Fourth quintile	Household furnishings	2378	2689	2261	2531	2593	2490.4
Canad	Average expenditure per household	Highest quintile	Total expenditure	143908	152452	156323	161780	164599	155812.4
Canad	Average expenditure per household	Highest quintile	Household furnishings	3693	3783	3494	3661	3761	3678.4

Next, we exported all the data from the geodatabase table demog (saved and attached as Matching demog to StasCan Quntiles.xlsx). Colums F1989 to F1999 provide the numbers of private households in each CDA by income levels in 2005 at $10,000 intervals. In the row “population”, we sum up number of households in each column, which represents the total population of each income level. In the column “accumulative population in 5 equal quintiles”, we first obtain the number of people

(34852) in each of the five quintiles by dividing the total population (174260) by 5, and then accumulatively add the number to itself to populate the column. Finally, we match groups of F19## columns to the five equal quintiles (as highlighted by matching colors in Table 2). Because the demog table divides population by income while the Statistics Canada table by equal population quintiles, we can only fit each number in "accumulative population in 5 equal quintile" to the closest value in "accumulative sum" from demog.

TABLE 2 ACCUMULATIVE POPULATION IN FIVE EQUAL QUINTILES

	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH
1	F1289	F1290	F1291	F1988	F1989	F1990	F1991	F1992	F1993	F1994	F1995	F1996	F1997	F1998	F1999
17	0	0	20	250	0	10	10	15	15	0	10	15	0	0	175
18	0	0	0	140	0	15	10	10	10	10	15	30	35	0	15
19	0	0	10	165	0	10	0	0	10	20	20	40	0	0	50
20	0	0	0	230	0	15	10	20	15	25	10	20	30	15	75
21	0	0	0	160	0	10	10	10	20	0	15	10	25	10	55
22	0	0	20	185	10	0	20	10	15	0	30	0	25	10	60
23	0	0	20	180	10	0	10	45	15	10	15	10	10	15	50
24				population	5430	12785	13910	16650	16410	14875	14985	13640	12450	10315	42810
25				accumulative sum	5430	18215	32125	48775	65185	80060	95045	108685	121135	131450	174260
26	matching Table 203-0022 quintiles				lowest quintile			second quintile		third quintile			forth quintile		highest quintile
27		average spending on furnishing	weight	accumulative population in 5 equal quintiles											
28		901.2	0.086716	34852											
29		1406.2	0.135308	69704											
30		1916.4	0.1844	104556											
31		2490.4	0.239632	139408											
32		3678.4	0.353944	174260											
33	sum	10392.6	1	174260											

We incorporate "average spending on furnishing" values from the "AVERAGE" column in Table 1. We may as well calculate the weights by dividing each "average spending on furnishing" value by the total, but we decide to use the exact expenditure numbers which directly indicate home hardware sales in each CDA despite appearing greater in number. Therefore, our conceptual equation to calculate demand should be:

$$\text{Demand} = population_{lowest\ quintile} * 901.2 + population_{second\ quintile} * 1409.2 + population_{third\ quintile} * 1916.4 + population_{fourth\ quintile} * 2490.4 + population_{fifth\ quintile} * 3678.4$$

In the Python code for Section 3.5, we create a string variable to store the demand formula below, and we call the function arcpy.CalculateField_management() using this string variable as a parameter.

```
demandCalculation = "((!demog.F1989! + !demog.F1990!
+ !demog.F1991!)*901.2)+(( !demog.F1992! + !demog.F1993!)*1409.2)
+(( !demog.F1994!+ !demog.F1995!+ !demog.F1996!)
*1916.4)+(( !demog.F1997!
+!demog.F1998!)*2490.4)+(( !demog.F1999!)*3678.4)"

arcpy.CalculateField_management("cdaView", "Demand", demandCalculation,
expression_type="PYTHON_9.3", code_block="")
```

In the Python script, two variables are created: cdaFilePath, and joinTable. The cdaFilePath variable points to the CDA feature class in RMoWFD, and the joinTable variable points to the demog table in the RMoW geodatabase. The arcpy.DeleteField_management variable deletes the Demand and Expenditure column if it exists within the CDA attributes table, which would only happen in this case if the script was run previously, otherwise it will do nothing on a brand new RMoW geodatabase. The two arcpy.AddField_management functions add the two Demand and Expenditure tables to the CDA feature class. The arcpy.MakeTableView_management function creates a table, cdaView, from an attributes table in a feature class, in this case CDA. The arcpy.AddJoin_management function combines the cdaView table with the joinTable demog table using "DAUID" as the combining column. The arcpy.CalculateField_management function calculates the demand using the demandCalculation parameter and adds it to the Demand column in cdaView, while arcpy.RemoveJoin_management disconnects the cdaView and demog tables.

VI. Creating Demand Points

A variable demandPoints points to the to be created demandPoints feature class. In arcpy.FeatureToPoint_management, it takes the cdaFilePath and demandPoints feature classes and combines them together, as the "INSIDE" parameter is active, which will create a points feature class demandPoints, and then place the demandPoints on each respective location within the CDA feature class.

VII. Running Location-Allocation Analysis

We begin with studying the sample code for running location-allocation, which is found at Tool Help in Network Analyst Tools -->Analysis --> Make Location-Allocation Layer. A prerequisite for performing Network Analyst is checking the extension by using the statement below:

```
arcpy.CheckOutExtension("Network")
```

We use the time.time() function at the beginning and the end of the chunk of code for each scenario, in order to print the running time to keep track of the process when running the code. For example, the code for scenario 1 begins with creating a string variable to record the strating time:

```
startTime37S1=time.time()
```

and ends with a print statement to show the time span:

```
print "3.7 Location-Allocation Scenario 1: "+ str(time.time()-startTime37S1)+" seconds
```

The first thing to do in location-allocation (L-A) is to create a new L-A layer. This can be done in Python by calling arcpy.na.MakeLocationAllocationLayer(), which is a function requiring 17 parameter (some optional). We make sure we assign all the parameters correctly by reading the Tool Help page and using proper indention to increase readability. Our network analysis is consistent in setting the following parameters: in_network_dataset andimpedanceAttribute (we use the same network dataset, and the field EstTravelTime as impedance), loc_alloc_from_to (from demand to facility, because we need to generate lines from each demand point to its server store), loc_alloc_problem_type (we use "MAXIMIZE_Market_Share" as our strategy based on demand in the CDAs), impedance_cutoff (17 minutes of drive), and the others as default as in performing L-A using ArcMap in Assignment 3. We agree with Marchant (2014) that the average time that a consumer is willing to travel to a local business is 17 minutes, based on over 800 responses across the United States with little variance among age groups. Major differences in code between the three scenarios include: names of the out_network_analysis_layer ("Scenario1", "Scenario2", and "Scenario3"), and number_facilities_to_find (12 for all the existing stores in Scenario 1, 1 for Scenario 2 to select one single site, and 2 for Scenario 3 to choose two optimal locations).

Next, we need to load store locations and demand points into the L-A layer. We begin with obtaining the names of the two sublayers in the L-A layer by creating a dictionary variable using the function arcpy.na.GetNAClassNames(), and then store the names "Facilities" and "DemandPoints" (elements in the dictionary variable) in string variables for later use. We then create a Python dictionary to set the default properties, and change the defaultValue for the ["FacilityType"] element to 0 (candidate), 1 (required), or 2 (competitor). We map the field name "Demand" to ["Weight"] as we based our analysis on demand, and we match the names so that the names can later be shown in the Network Analyst window. To load store locations into the Facilities layer, we call the arcpy.na.AddLocations() function with parameters set properly: input and output information is stored in string variables defined above; search tolerance as default; append = "APPEND" because we need to add candidates and competitors into one layer for Scenario 2 and 3; exclude_restricted_elements = "EXCLUDE" to prevent selecting locations that cannot be reached due to restrictions or barriers. Demand points are loaded in a similar procedure while setting the defaultValue is not relevant.

Finally, we call arcpy.na.Solve(), which is equivalent to clicking the Solve button in ArcMap, and save the layer to file using different names.We now move on to examine the results in ArcMap by adjusting the color scheme and performing the following analysis.

Results

I. The Three Scenarios

i. Scenario 1

Each store will be referred to in the format of Name ##, where Name is the name of the store, and ## is the ObjectID of the store in each respective scenario. In scenario 1, the base location-allocation analysis, Rona 13, on 730 Ottawa St. S., Kitchener, Ontario, has the most demand weight, 37295429.67, being 18% higher than the mean, 30645638.75, and 52% higher than the minimum demand weight, 19346055.83, while Home Depot 17, on 100 Gateway Park Dr., Kitchener, Ontario, the store with the second highest DemandWeight, 36607007.26 (2% lower than Rona 13), has the higher amount of DemandPoints allocated to the store, at 642 vs 623. When observing the below, in Figure 5, the Rona 13 store is in a better location distance-wise from the Home Depot 17 store in Figure 6. Since the Rona store is in a better location, it can be considered to be the store that delivers demand in the most efficient way.

FIGURE 5 RONA

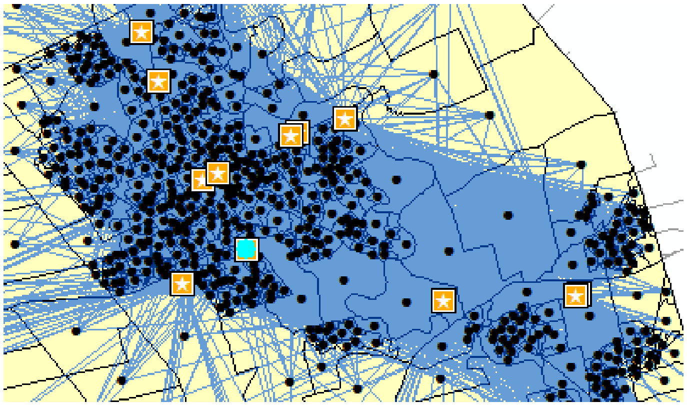
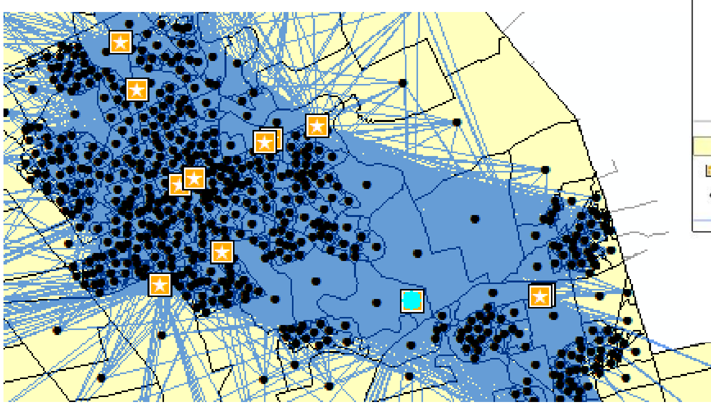
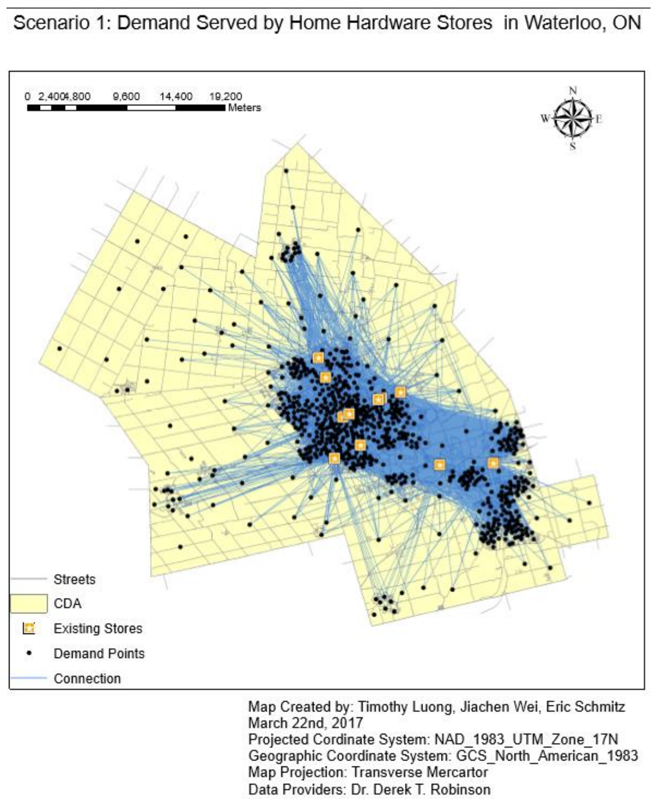


FIGURE 6 HOME DEPOT

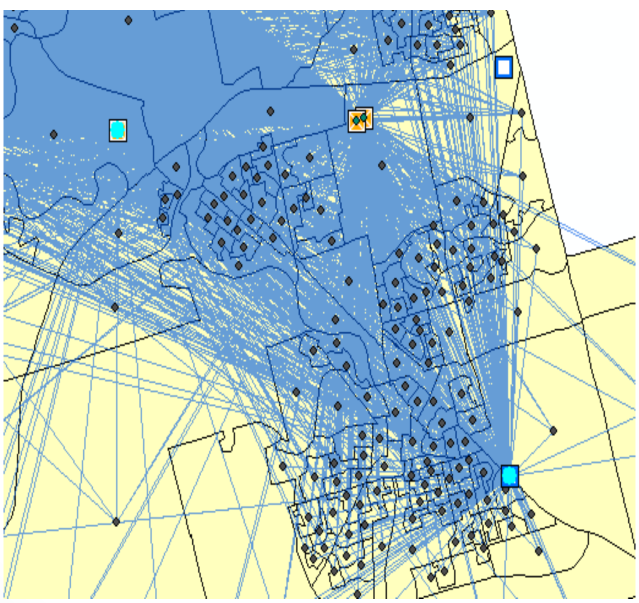


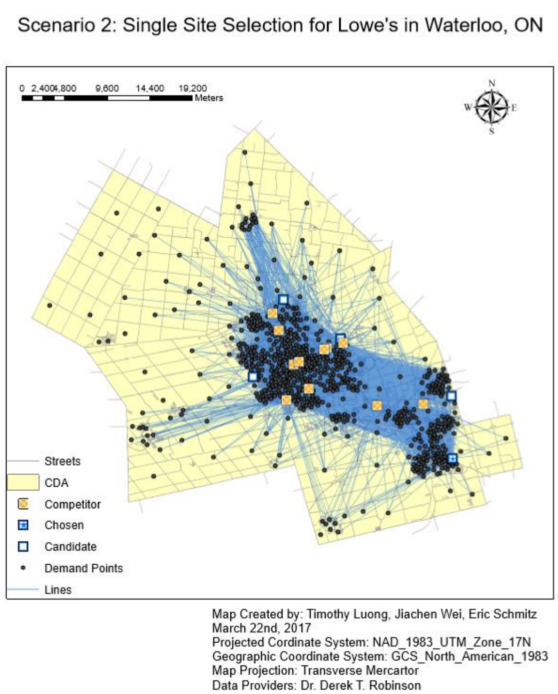


ii. Scenario 2

In scenario 2, the single site selection analysis, one Lowe’s store was selected to be the most preferred location amongst 4 other candidate store locations. This is Lowe’s 05, which is the Dundas Street South location in Cambridge, Ontario. Comparing the Lowe’s to the other stores, Rona 42 remains the that services the most demand in the DemandWeight column. However, the mean DemandWeight has gone down to 27668400.15. Therefore, Rona 42 (with a DemandWeight of 34670857.60, 7% decrease from Scenario 1), on 730 Ottawa St. S., Kitchener, Ontario, is 20% higher than the mean DemandWeight. This time, the store with the second highest DemandWeight is Canadian Tire 52, which is the store on 417 King St W. Kitchener, Ontario. The Home Depot from scenario 1 has become the 5th highest store in terms of DemandWeight (29101677.26, 5% higher than the mean, 21% decrease from Scenario 1), but remains almost the same in terms of DemandCount (641). This is because the Lowe's 05 store has taken a large portion of the demand points belonging to Home Depot 46, on 100 Gateway Park Dr., Kitchener, Ontario, illustrated by Figure 7. In fact, Lowe's 05 is the 6th highest store in terms of DemandWeight (29069448.46), but has the Lowe'st DemandCount of 243. The store with the Lowe'st DemandWeight served (18693928.76, 32% lower than the mean, 3% decrease from Scenario 1) is the Rona 53 store, which is on 5 Forwell Rd, Kitchener, Ontario. This location also happens to be one of the locations added in the a4_stores.csv file.

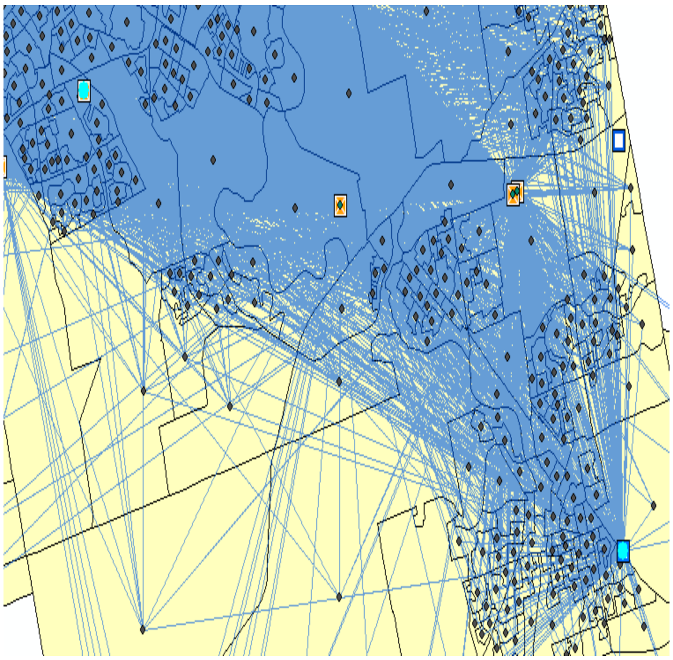
FIGURE 5 HOME DEPOT (TOP LEFT) AND LOWE'S BOTTOM RIGHT

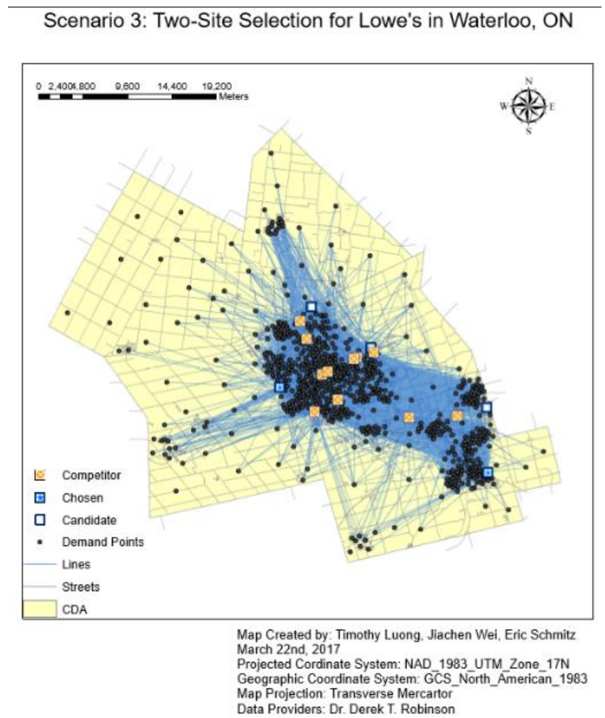




iii. Scenario 3

FIGURE 6 RONA (TOP LEFT) AND LOWE'S BOTTOM RIGHT





In scenario 3, the two-site selection analysis, two L owes stores were chosen for this analysis: Lowes 02 (which has a DemandWeight of 27214876.91), which is the Ira Needles location on 345 The Boardwalk, Waterloo, Ontario, and Lowes 05 (which has a DemandWeight of 28978885.81), which is still the Dundas Street S location in Cambridge, Ontario. The mean DemandWeight is (25914634.14). The store with the highest DemandWeight (31387222.75, 17% higher than the mean, 9% decrease from Scenario 2, 16% decrease from Scenario 1) served remains the Rona 18 store, which is on 730 Ottawa St. S., Kitchener, Ontario. The store with the least demand served (17332107.04, 33% lower than the mean, a 7% decrease from Scenario 2, 10% decrease from Scenario 1) is still the Rona 29 store, on 5 Forwell Rd, Kitchener, Ontario. In this scenario, Rona 18 and Lowes 05 has a minimal or no effect on each other in terms of DemandCount or DemandWeight, because these stores are further apart from each other, and do not draw much of their demand from their opposing store's city, Kitchener and Cambridge, illustrated in Figure8.

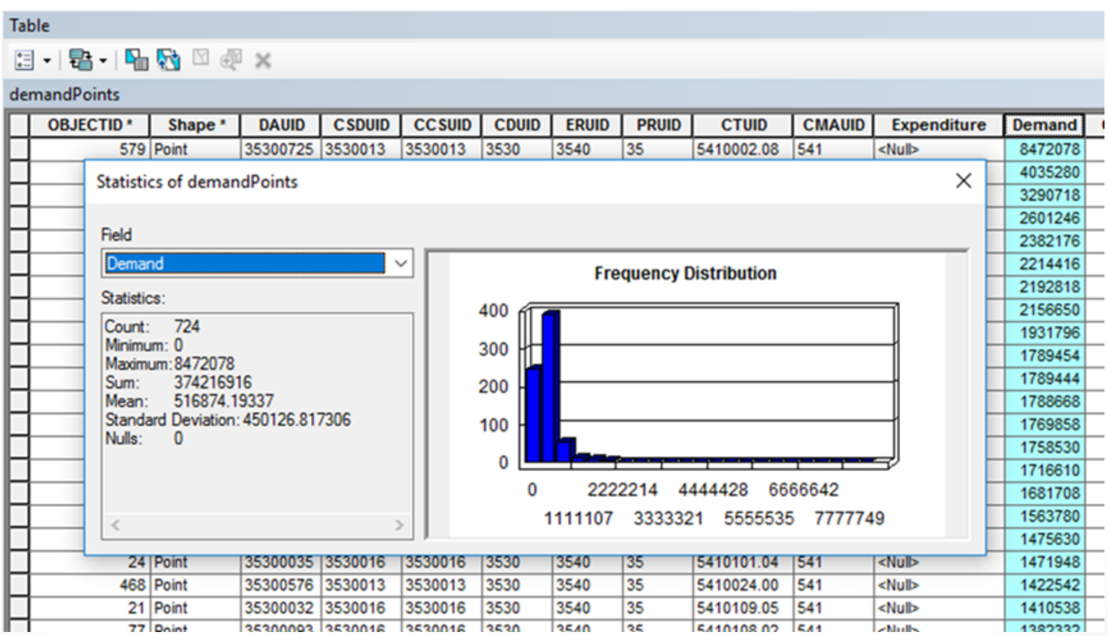
II. Viability of the New Building Centre(s)

The two prime locations for Lowe's building centres are on Ira Needles Boulevard in Waterloo, and South Dundas street in Cambridge. Firstly, the Ira Needles Boulevard location is in an extremely special location for shoppers and traffic. It is unique from the other locations in the way that this the area has other big box retail stores that do not only involve home hardware. This is because Ira Needles Boulevard in Waterloo is home to something called The Boardwalk. The Boardwalk is a massive, new big box retail location packed with different types of stores, services, and restaurants. This adds a lot of value to this location because of how many different types of people will flock to The Boardwalk to shop (Biba 2006). Biba found in a study that more people today are inclined to go to a big box retail location rather than spend time driving to stores located in different parts of the city. Leszczyc (2004) also found in a similar study that grocery shoppers in particular are more likely to do their grocery shopping in an area that has the other stores they need to visit in that area. The variation of types of stores in the area makes it a hotspot for shoppers. Not all of these people may be coming exclusively to shop at Lowe's, so the added chance of impulse buyers who could decide to stop at Lowe's makes this a great location. Besides the boardwalk Ira Needles Boulevard is also the main road out of Waterloo and towards Stratford and Western Ontario. We can assume heavy traffic coming down this road constantly. Another top location for a Lowe's improvement store is in southeastern Cambridge on Dundas St S. This store is near downtown Cambridge, surrounded by residential areas with lots of activity. A study performed by Kumar (2000) found that “sales of a grocery store are positively related to the total number of households in the trade area”. This is because a larger number of households in the area results in heavier traffic to the store. The YMCA is just up the road, as well as many restaurants. These kinds of services around a store make it a high possibility that an individual is more likely to make a trip to Lowe’s while they do their daily chores around town. A downfall of this location can be the amount of turns cars have to make on residential roads, as well as possible traffic down the large Coronation Blvd.

This section examines three aspects of validity for the selected new locations. One, the locations should cater to the high-demand CDAs; two, the analysis should not be subject to a high sensitivity; three, speed limit of the neighboring streets. This section investigates validity from the three perspectives using the result of the two-location scenario (as this result show the two most desirable locations together).

i. Response to Demand

FIGURE 9 STATISTICS OF DEMAND COLUMN



To analyze how well the selected locations satisfy the local demand, we can create a summary circles of the demand points. First, we need to select the points with "high demands"—demand values above a moderate level. In Figure 9, the statistics window shows a skewed distribution, which suggests that the easily obtained mean value may not well represent the middle demand level, and thus is not optimal for our purpose. This is confirmed when we only filter out mere 210 out of the total 724 demand point using the following SQL statement:

SELECT * FROM demandPoints WHERE Demand >= 516874.19337

We export the attribute table to a txt file, and copy the data into Excel to calculate the median for the Demand column, which is 423120 Canadian dollars. As expected, half of the points (362 out of 724) are selected as "high demand" points using the statement below:

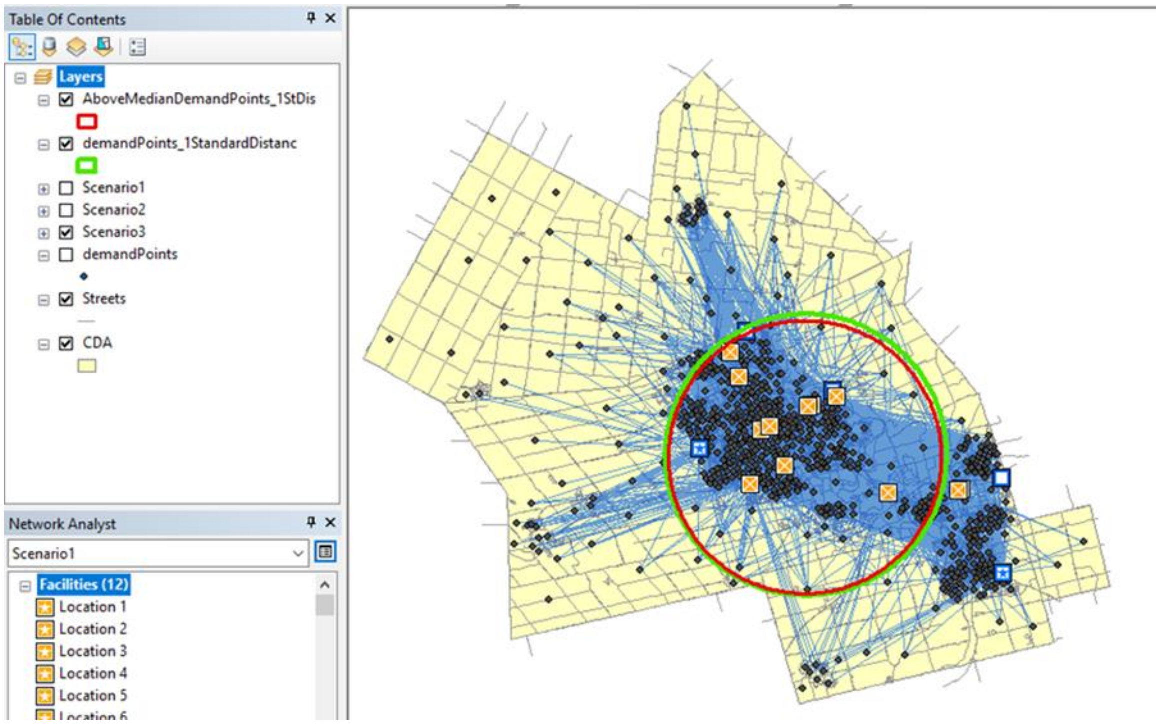
SELECT * FROM demandPoints WHERE Demand >=423120

We hereby create a point feature class using Add XY Coordinates, and draw two circles using the Standard Distance tool. The Standard Distance tool calculates the mean X, Y coordinates of the input feature class, and creates a circle with a specified radius (1 standard distance, 2 standard distance, or 3 standard distance). Standard distance (d), defined by the equation below, is the spatial equivalent to standard deviation in basic statistics.

$$d = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu_x)^2 + (x_i - \mu_y)^2}{n}}$$

Figure 10 shows the 1d summary circle (green) of all the demand pointsand the 1d summary circle (red) of the high-demand points. The center of a summary circle is the mean X, Y values of the demand points. The red circle moves slightly towards the northwest from the red circle. This seem to suggest the location on Ira Needles Boulevard is closer to the high-demand districts, as it is located closer to the center of the high-demand circle (red) than to the center of the all-demand circle. Depending on how demand is calculated, the movement of the summary circle for all CDAs and above-median CDAs can give meaningful feedback on which location caters to higher demand (but this is not applicable to our result since the two circles are almost overlapping). Contrary to the result in Scenario 2, the Ira Needles seems an overall better location than the other site.

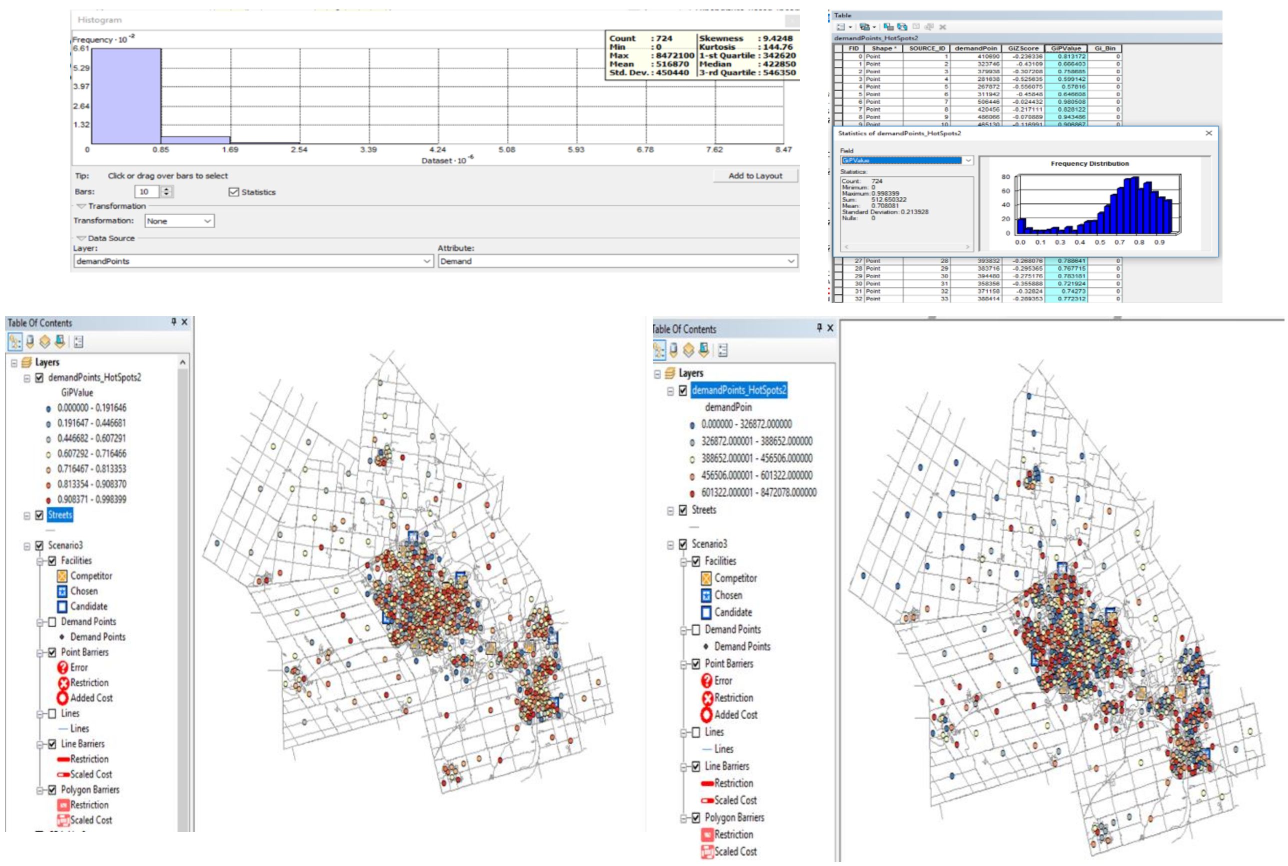
FIGURE 10 SUMMARY CIRCLES



Next, we run a hot spot analysis using all the demand points. The Hot Spot Analysis tool in ArcGIS looks at the neighboring features, and considers a feature as a statistically significant hot spot only if it is surrounded by high-value features

(ArcGIS Pro, 2017 a). We set the parameter Conceptualization_of_Spatial_Relationships as INVERSE_DISTANCE, which draws greater influence from the neighboring features than from the distant features, because we believe that the demand for home improvement is spatially autocorrelated.

FIGURE 11 HOT SPOT ANALYSIS

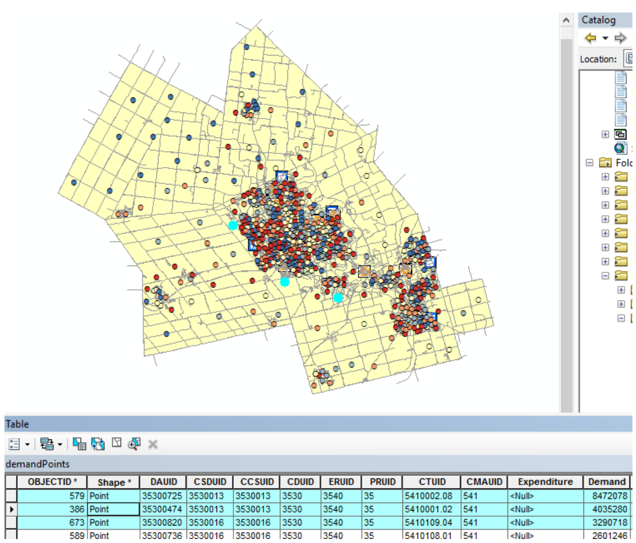


The Hot Spot Analysis tool returns a z-score and a p-value along with each feature's Gi* value (ArcGIS Pro, 2017 a). The p-value is the probability indicating confidence levels, and the most intense concentrations (hot spots) occur where the p values are the smallest (ArcGIS Pro, 2017 b). Figure 11 (lower left) is the p-value rendering result, classified into 7 classes using Natural Breaks (Jenks) to maximize the differences between classes (ArcGIS Pro, 2017 c). We choose Natural Breaks over the confidence level breaks (±1.65, ±1.96, and ±2.58) because the histogram indicates an extremely skewed distribution, and the p-values are predominantly greater than 0.5 (very low confidence). We use another rendering showing Demand values by 5 quintiles (Figure 11 lower right), both results confirm that the selected locations makes economic sense as they are in the vicinity of a significantly greater number of high-demand points (rendered as hot spots in our approaches) while staying relatively far away from competitors.

ii. Sensitivity Analysis

We examine how sensitive our approach is to a subtle change. If a different location is suggested after removing a small number of demand points (representing a small decrease in population, or saturation of demand in these CDAs), then our analysis may not be strong or reliable enough. The first step is to delete the three objects with the highest Demand values using Editor (when the Demand column is in descending order). Figure 12 shows the three selected objects to be deleted. After

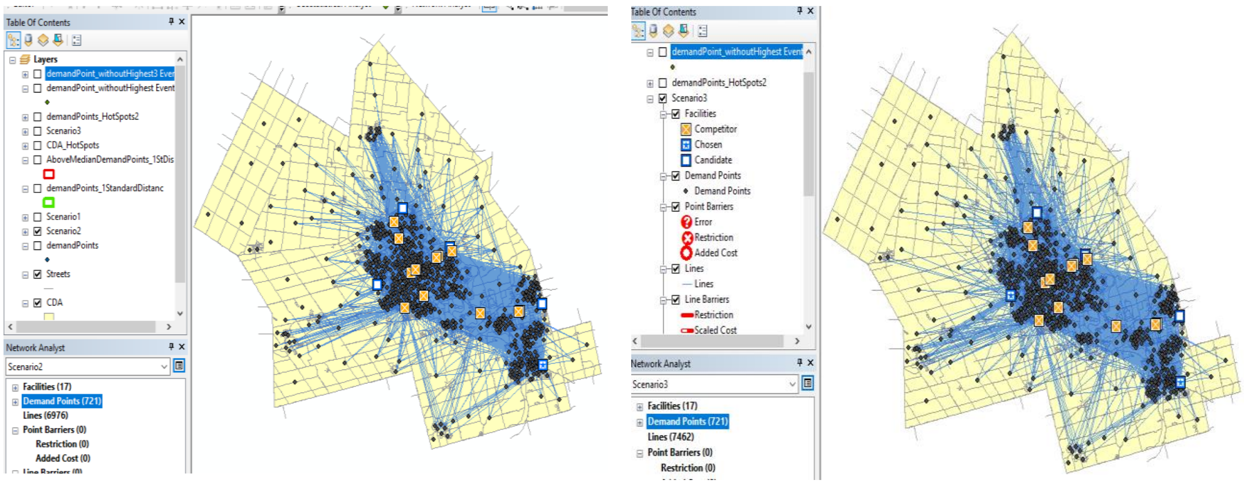
FIGURE 12 DELETING 3 POINTS



OBJECTID *	Shape *	DAUID	CSDUID	CCSUID	CDUID	ERUID	PRUID	CTUID	CMAUID	Expenditure	Demand
579	Point	35300725	3530013	3530013	3530	3540	35	5410002.08	541	<Null>	8472078
386	Point	35300474	3530013	3530013	3530	3540	35	5410001.02	541	<Null>	4035280
673	Point	35300820	3530016	3530016	3530	3540	35	5410109.04	541	<Null>	3290718
589	Point	35300736	3530016	3530016	3530	3540	35	5410108.01	541	<Null>	2601246

that we display the attribute table as a point feature class (right click the layer and go to the option name Display XY Data), and loaded the points into the network analysis window as Demand Points. From here we solve it again for Scenario 2 and 3, and the result is the same as before when we have the entire 724 demand points, (Figure 13). This identical result indicates that we have a sound analysis which is not overly sensitive to change.

FIGURE 13 SENSITIVITY ANALYSIS FOR SCENARIO 2(LEFT) AND 3(RIGHT)

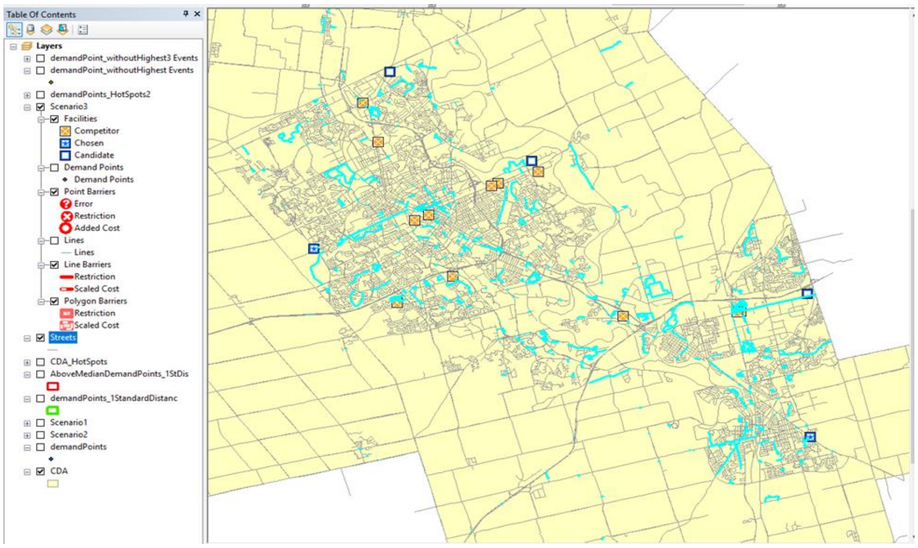


iii. Slow Traffic

In addition to demand level, we also consider speed limit of the neighboring streets to make sure the store is not located close to special places like schools. We simply examine this by selecting the streets with the SQL statement below, where 40 km/h is the speed limit in school zones I Ontario. The result (Figure 14) seems to suggest that traffic speed is not a concern for both of our selected sites, which are located away from slow traffic.

SELECT * FROM Streets WHERE Speed≤40

FIGURE 14 SLOW STREETS



III. Impact of the New Centre(s) on Existing Building Centres

As the more Lowes stores were considered as candidates for building, the more the demand served by the competitor stores decreased in tandem. The order in which the stores served demand did not change too much, but the store that the Lowes were closest did in fact become greatly affected, as in Scenario 2, where the second highest demand serving store became the fifth. The lowes did not become a large competitor to the already existing stores until Scenario 3, where the Lowes store in Cambridge, Ontario became the second highest demand serving store. It seems like it would be unlikely for a Lowes to become the highest store in terms of demand served due to the dividing of DemandWeight. If a Lowes store was placed closed to the Rona that has the most demand served in all 3 scenarios, neither store would become the highest or close to high in terms of demand served, evidenced by Scenario 2, where competition with the second highest store brought both stores down to the middle in terms of demand served.

Discussion

This report is made from different types of data used together in the way we believe proves to Lowe's what achieves the highest amount of demand for store locations in the Waterloo region. We believe that drive time in minutes to the store location is the most important factor when it comes to choosing one store over another. Although, different data sets can be used to get a different result when it comes to what store location is the optimal site. Non linear data that can be useful to determining store location are things such as unemployment rates, tax rates, housing market information, etc. Some of these other data sets and their implications will be discussed below.

I. Disposable Income

For an individual to go out and spend money at a home improvement store, they need to have disposable income. A consumer with a good job and high income would be willing to make the trip out to a store more often than one without a job. Personal consumption expenditure makes up about 70% of the GDP. Employment and income are some of the most prominent variables affecting personal consumption expenditure (Phalguni, 2015). People with more disposable income are more likely to be an impulse shopper and spend money at stores for other psychological reasons (Tauber, 1972). Tauber found that shoppers fall into different psychological types. The types of mindsets most associated with disposable income are the diversion shopper, self-gratification, sensory stimulation, and status and authority shopping. Without disposable income these shoppers would not be as enabled to spend money. With this information we can assume that a store closer to higher income neighborhoods will have a higher chance to be more profitable and popular. This kind of information is very practical because it can also help dictate the price of goods in the store. It is hard information to collect and map though. Unemployment rates and income should be averaged out and assigned to neighborhoods, but there is lots of variation.

II. Tax Rate

Also related to employment status and income is the tax rate individuals pay. No matter the income, every individual will have to pay some amount of taxes. The higher the taxes, the less money the consumer has left to spend at a home improvement store. Ontario has very high hydro bills, and people living in more rural Ontario can have hydro bills 3x more expensive than those in the city near the grid (Phalguni, 2015). With this information we can assume that individuals coming from rural settings or areas with higher taxes, will be less likely to travel to the home improvement store. It is important to understand all of these extra costs. Looking at household expenditure may not give the entire story of the household.

III. Housing Market

When it comes to finding the demand for home improvement stores, it is very helpful to look at one of the largest factors of people buying these types of goods; the housing market. There are three types of customers at home improvement stores. There is the do-it-yourself customer who buys the products and does the work themselves. In contrast to that there is the do-it-for-me customer who purchases the products, and the option for a third party to do the work for them. Finally, there is the professional customer who could be a contractor or other type of company making a larger purchase (Phalguni, 2015). Understanding how the housing market and these types of customers go together is important in understanding the type and size of crowd coming to the store. For example, there is data available that tracks the amount of housing starts and permits being issued. This means an influx of all three types of customers who are involved in construction projects around the house, either big or small. With this information we can assume that areas with more construction and building permits being issued will be more likely to go to home improvement stores. A study conducted by Wilder (1988) found that "while increases or decreases in sales of new homes affect industry output, sales of previously occupied homes have an even greater impact". It also went on to state that "during periods of economic expansion, hardware items are purchased to prepare homes for resale, to remodel older homes, and to maintain new and existing homes". This means that with an increase in construction and economic expansion, hardware stores in the area can expect higher demand and sales. Waterloo is a city filled with construction and expansion. It is a good location because of how many apartments are being built as well as residential houses that are demolished and replaced with newer styles of housing. There is currently a high amount of professional types of customers in the area. In addition, the old age of the houses in the area means that more maintenance will constantly be required. This is a good thing for home improvement stores. Being closer to areas of construction and housing projects means a higher demand for the store location.

IV. Limitations

With all this data available, there are still limitations. Our data is from the Average household expenditure, by province (Ontario) table from Stats Canada. We specifically pulled from the expenditure category household furnishings and equipment to understand the average amount a household spends on furnishings and equipment. To get a more accurate result we organized the category into age groups. Some of the limitations we ran into with this data was outdated demographics, no adjustment for inflation in household expenditure, only using statistics on age, and external factors.

The data we pulled from could be relatively inaccurate because of its age. The most recent year the data was published in 2015. From 2014 to 2015 the amount of expenditure in the category household furnishings and equipment dropped by 90$ (Canada, 2017). This is not a huge change but it is hard to say how much would have changed by 2017, and if there would have been a bigger change to the difference between age groups. The data also doesn't add up when going through population size. 100% of the population is somehow two less than 100% of the total population by sex and age groups. This does not make a large difference at all, but can lead to errors.

Another limitation that directly affects the dollar amount in the table is how there is no adjustment for inflation in household expenditure. Because of rising prices, the amount of expenditure may not be as closely related to willing buyers as it is to furniture that is more expensive. We are looking at the dollar amount as an indicator of how often people are spending their money at a home improvement store. We do not know what products the consumer is spending their money on. If it were a one-time purchase of a large piece of furniture, this would skew the data. This is especially true if furniture is becoming more expensive. Some consumers may fall into this category of most of their average expenditure coming from one or two purchases while others may make many trips. It is very hard to tell the difference between these groups. The survey for household expenditure would have to involve what the individual spent their money on.

One of our largest limitations happened to be set by ourselves. We only considered statistics on age and income. Other data such as gender, education, and marital status were considered but not used because of the difficulty to implement. These types of data are very subjective and variable. In order to maintain accuracy we stuck to more simple data use. It has already been discussed how it is our belief that areas with a higher average income would most likely make more trips to the home improvement store. It is hard to say if gender, education, and marriage would affect how often an individual goes to a certain location.

The external factors that act as limitations are things that need to be thought of outside of the network analysis. These types of things involve traffic, parking, other types of stores around the location, and difference in 17 minutes of driving in our network dataset versus 17 minutes of driving in consumer's perception. These limitations prove that real life can be very different to our simulation in ArcGIS and there are certain things that need to be accounted for. Another limitation comes with the size of these stores. Large stores like these can only hold so many people. If there are not enough checkout counters to handle the flow of people then sales could be sacrificed. A study by Kumar (2000) found that "sales per square foot were higher for stores with higher number of checkout counters per square foot of store area" in grocery stores. Grocery stores can be seen as a similar example due to the size and traffic flow. Without adequate services inside to handle heavy demand the store becomes limited to a certain sales per square foot. Other things such as parking, traffic, and real world driving time are changing all the time and very hard to account for. What we can account for is that being close to things such as large parking lots, and highways may help the driving time and parking factors.

V. Store Longevity

Retail stores need to be built to stand the test of time, and to achieve this the environment of the retail store must be considered. Waterloo is a rapidly growing city; areas that may not be very built up now may be highly populated in the coming years. A study performed by Harris and Ullman (1945) found that the growth of retail and commerce is directly related to the growth of the city they reside in. The city is comprised of different types of zones that interact and support each other economically (Harris, 1945). To account for this it is important to look at the area surrounding the store location to confirm there is room for city expansion around the store. It is also important to be close to things such as highways since these are always a steady flow of traffic. Something else that needs to be considered is the aesthetics of the store.

> "Consumer's beliefs about the physical attractiveness of a store had a higher correlation with patronage intentions than did merchandise quality, general price level, selection, and six other store/product beliefs" (Baker, 1992).

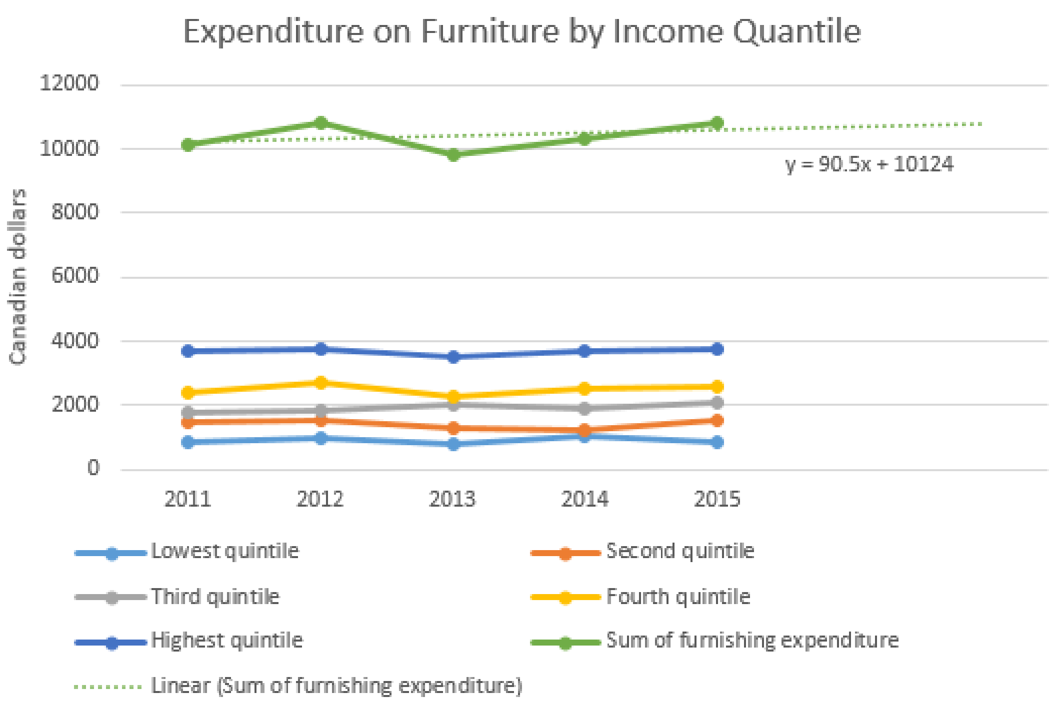
This quote from Baker's (1992) paper displays the importance of maintaining physical attractiveness throughout the life of the store. A retail store that remains open for many years must keep a modern appearance that is pleasing to the customer to maintain demand (Doyle, 2007). In Doyle's (2007) study he recommends using cheaper building materials if "the lifespan of the design is likely to be brief". This is because constant upgrading and modernizing will be required, and minimizing costs is important to longevity. This was proven in the 1990s when luxury department store, Neiman Marcus spent $200 million within five years to renovate only 23 stores (Baker 1992).

VI. Expenditure Trends and Inflation

The hardware industry has been a growing industry in the previous years. The growth rate in the hardware store industry rose at an average annual rate of 1.3% from 1972 to 1986 (Wilder, 1988). As you can see in Figure 15, there is a linear trend in expenditure on furniture and home improvements goods increasing in Canada since 2011 (Canada, 2017). The growth rate in expenditure on household goods from 2011 to 2015 is 2.6% (refer to Expenditure_Based_Income.csv).

Growth Rate(%) = 100* (the year after - the year before) / the year after

FIGURE 15 UPWARD TREND IN EXPENDITURE ON FURNITURE



Compared to inflation between 2005 to 2015 of 18.32%, or an average of 1.7% per year (Inflation Calculator, 2017). This means that people are spending more on home improvement goods. If the trend of home improvement industry growth, expenditure, and inflation continue then Lowe's can expect high demand and spending. This confirms an increasing demand for household furnishings and home improvement equipment, as well as healthy price inflation. If the percentage of expenditure stays above inflation, this is a good sign for home improvement stores. As Waterloo continues to grow and bring in new housing projects such as discussed above, this will increase demand for building center supplies in the years to come.

References

ArcGIS Pro. (2017 a). How Hot Spot Analysis (Getis-Ord Gi*) works. Retrieved from http://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/h-how-hot-spot-analysis-getis-ord-gi-spatial-stati.htm

ArcGIS Pro. (2017 b). What is a z-score? What is a p-value? Retrieved from http://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/what-is-a-z-score-what-is-a-p-value.htm

ArcGIS Pro. (2017 c). Data classification methods. Retrieved from http://pro.arcgis.com/en/pro-app/help/mapping/symbols-and-styles/data-classification-methods.htm

Baker, J., Levy, M., & Grewal, D. (1992). An experimental approach to making retail store environmental decisions. Journal of retailing, 68(4), 445.

Biba, G., Rosiers, F. D., Theriault, M., & Villeneuve, P. (2006). International articles: Big boxes versus traditional shopping centers: Looking at households' shopping trip patterns. Journal of Real Estate Literature, 14(2), 175-182,187-188,190-192,195-202.

Canada, G. O. (2017, January 27). Average household expenditure, by province (Ontario). Retrieved March 06, 2017, from http://www.statcan.gc.ca/tables-tableaux/sum-som/l01/cst01/famil130g-eng.htm

Doyle, S. (2007). Retail store design. International Retail Marketing, 78.

Harris, C. D., & Ullman, E. L. (1945). The nature of cities. The Annals of the American Academy of Political and Social Science, 242(1), 7-17.

Inflation Calculator. (n.d.). Retrieved March 22, 2017, from http://www.bankofcanada.ca/rates/related/inflation-calculator/

Kumar, V., & Karande, K. (2000). The effect of retail store environment on retailer performance. Journal of Business Research, 49(2), 167-181.

Leszczyc, P. T., Sinha, A., & Sahgal, A. (2004). The effect of multi-purpose shopping on pricing and location strategy for grocery stores. Journal of Retailing, 80(2), 85-99. doi:10.1016/j.jretai.2004.04.006

Phalguni Soni - Disclosure Mar 24, 2015 4:11 pm EST. (n.d.). Home Depot's Product Portfolio and Supplier Relationships. Retrieved March 06, 2017, from http://marketrealist.com/2015/03/home-depots-product-portfolio-supplier-relationships/

Ross Marchant. (2014). Consumers will Travel 17 Minutes to Reach a Local Business. Retrieved from https://www.brightlocal.com/2014/05/01/local-business-travel-times/

Tauber, E. (1972). Why Do People Shop? Journal of Marketing, 36(4), 46-49. doi:10.2307/1250426

Wilder, P. S., & Klarquist, V. L. (1988). Retail hardware stores register productivity gain. Monthly Labor Review, 111(5), 39-43.

Appendix

Appendix 1: Python Script Explanation

This section will provide a quick description of sections of code in the Python script. More detail is provided in Section 2 of this report.

Code Section 3.3

This section geocodes competitor stores by taking in a csv file, referencing it to a list of addresses provided by ArcGIS, and maps these stores to an Address Locator feature class, allowing it to be interpreted and manipulated geographically by ArcGIS/Arcpy functions.

Code Section 3.4

This section merges the competitor stores output by code section 3.3 into the existing stores that have been provided for the purposes of this analysis. It ensures that the new stores matches the feature object format that currently exists in the provided data by removing unneeded field columns, and then overwriting the old collection of stores with this new collection.

Code Section 3.5

This section calculates an estimation of demand from each Census Dissemination Area (CDA) based on the average household expenditure per quintile compared to the average Canadian household expenditure on household furnishings and equipment and/or maintenance.

Code Section 3.6

This section turns each CDA polygon into a demand point, which will be used for the network analysis.

Root

A root path variable will be created using os.path.dirname(sys.argv[0]). Os.path.dirname returns the address of which the Python script is being run from, while sys.argv[0] will return this value as a string to be assigned to the variable "root".

Env.workspace

An environment workspace created using the root path and appending "\\RMOW.gdb". This points to the RMOW geodatabase folder that is included in the submission.

Arcpy.env.overwriteOutput

Allows any arcpy functions to overwrite duplicate entities that exist within the gdb folder.

Network

The file directory for the Network Dataset: env.workspace + '\\RMoWFD\\RMoWFD_ND'

impedanceAttribute

This is the value "EstTravelTime", which is what is considered an impedance to travelling to a certain store.

Oneway_restriction

This is a value which shows if a street is a one-way street or not.

outLayerFile_S1

Output layer file directory for Scenario 1: root + "\\Scenario1"

outLayerFile_S2

Output layer file directory for Scenario 2: root + "\\Scenario2"

outLayerFile_S3

Output layer file directory for Scenario 3: root + "\\Scenario3"

Appendix 2: Full Python Code

```
# # -*- coding: utf-8 -*-

# ---------------------------------------------------------------------------

# Author: Eric Schmitz, Jiachen Wei, and Timothy Luong  (Group 3)
```

Date Created: 18 March 2017

Acknowledgement: Dr. Derek T. Robison

Purpose: GEOG Winter 2017 Assignment 4

Using Network Analyst to perform location allocation for different scenarios.

Usage: CDA, ExisitngBldgCenters, PotentialSites,newStores, Municipalities, streets, demog, demog_fieldlookup, RMoWFD_ND,RMoWFD_ND_Junctions

Preconditions:

1) The network dataset (RMoWFD_ND) has been created manually using GUI

2) This .py file is stored in the folder containing RMOW.gdb

Postconditions:

1) candidates geocoded; competitors (existing) merged

2) demand calculated and represented as point features

3) three scenarios addressed

--

Import libraries and set the environment

import os, arcpy, sys

```
from arcpy import env


 # 3.2 setting relative path (according to Preconditions)

root = os.path.dirname(sys.argv[0])

env.workspace = root + "\\RMOW.gdb"

arcpy.env.overwriteOutput = True   #enabling overwrite


# Set parameters for 3.7 MakeLocationAllocationLayer()

network = env.workspace + '\\RMoWFD\\RMoWFD_ND'

impedanceAttribute = "EstTravelTime"  #exactly as the field name

Oneway_restriction = "OneWay"

outLayerFile_S1 = root +  "\\Scenario1"

outLayerFile_S2 = root +  "\\Scenario2"

outLayerFile_S3 = root +  "\\Scenario3"


#relative path to be used in 3.3 and 3.7
```

origStores = env.workspace + "\\RMoWFD\\ExistingBLDGCentres" #original stores are in this layer

candidateStores = env.workspace + "\\RMoWFD\\PotentialSites" #candidate stores are in this layer

#3.3 geocode additional competitor stores

startTime33=time.time()

storesToGeocode = root + "\\a4_stores.csv" # a string variable storing the path to the csv file of all competitors

#fieldMap is a string to be used as a parameter later (describing how fields in the network layer will map to the AddresLocator attribute fields)

#Use CreateAddressLocator_geocoding() to to create an object that can geocode locations

#CreateAddressLocator_geocoding (in_address_locator_style, in_reference_data, in_field_map, out_address_locator, {config_keyword}, {enable_suggestions})

fieldMap = "'Feature ID' OBJECTID VISIBLE NONE;'*From Left' L_FIRST VISIBLE NONE;'*To Left' L_LAST VISIBLE NONE;'*From Right' R_FIRST VISIBLE NONE;'*To Right' R_LAST VISIBLE NONE;'Prefix Direction' <None> VISIBLE NONE;'Prefix Type' <None> VISIBLE NONE;'*Street Name' FULL_NAME VISIBLE NONE;'Suffix Type' <None> VISIBLE

```
NONE;'Suffix Direction' <None> VISIBLE NONE;'Left City or Place' L_MUNIC VISIBLE NONE;'Right City or Place' R_MUNIC VISIBLE NONE;'Left ZIP Code' <None> VISIBLE NONE;'Right ZIP Code' <None> VISIBLE NONE;'Left State' <None> VISIBLE NONE;'Right State' <None> VISIBLE NONE;'Left Street ID' <None> VISIBLE NONE;'Right Street ID' <None> VISIBLE NONE;'Display X' <None> VISIBLE NONE;'Display Y' <None> VISIBLE NONE;'Min X value for extent' <None> VISIBLE NONE;'Max X value for extent' <None> VISIBLE NONE;'Min Y value for extent' <None> VISIBLE NONE;'Max Y value for extent' <None> VISIBLE NONE;'Left parity' <None> VISIBLE NONE;'Right parity' <None> VISIBLE NONE;'Left Additional Field' <None> VISIBLE NONE;'Right Additional Field' <None> VISIBLE NONE;'Altname JoinID' <None> VISIBLE NONE"

locator_style = "US Address - Dual Ranges"

reference_data = "'" + env.workspace + "\\RMoWFD\streets' 'Primary Table'"

addressLocator = env.workspace + "\\addressLocator"

arcpy.CreateAddressLocator_geocoding(locator_style, reference_data, fieldMap,
    addressLocator, "", "DISABLED")

outAddresses = env.workspace + "\\newStores"

arcpy.GeocodeAddresses_geocoding(storesToGeocode, addressLocator, "Street
    Address VISIBLE NONE;City Municipality VISIBLE NONE", outAddresses,
    "STATIC")
```

```
print "3.3 Geocode Additional Competitor Stores: "+ str(time.time()-startTime33)+" seconds"

#3.4 merging existing and new stores

startTime34=time.time()

#merge new competitors (assig4_stores.xlsx) with the existing points stored in the “ExistingBldgCentres” data layer

#Merge() in the code is alternative to using Data Management Tools --> Merge

#Merge_management (inputs, output, {field_mappings}) combines multiple input datasets of the same data type into a new output dataset

fields = arcpy.ListFields(outAddresses)

fieldNames = map(lambda n: n.name, fields)

for name in ["ObjectID", "Shape", "Name", "Address", "Municipality", "Status", "Size"]:

    fieldNames.remove(name)  #erase all field names
```

```
arcpy.DeleteField_management(outAddresses, fieldNames)  #delete unneeded fields

outStorestmp = env.workspace + "\\RMoWFD\\ExistingBldgCentres_tmp"

arcpy.Merge_management([origStores, outAddresses], outStorestmp)

arcpy.Delete_management(origStores)  #to permanently delete data from disk

arcpy.Rename_management(outStorestmp, origStores)  #rename the output dataset

print "3.4 Merging Existing and New Stores: "+ str(time.time()-startTime34)+" seconds"
```

```
#3.5 demand estimation

startTime35=time.time()

cdaFilePath = env.workspace + "\\RMoWFD\CDA"  #a string variable storing the directory path to your census dissemination data

#add new fields

#AddField_management (in_table, field_name, field_type, {field_precision}, {field_scale}, {field_length}, {field_alias}, {field_is_nullable}, {field_is_required}, {field_domain})
```

```
arcpy.DeleteField_management(cdaFilePath,["Expenditure","Demand"]) #use this
    line when these 2 fields exist

arcpy.AddField_management(cdaFilePath, "Expenditure", "DOUBLE", 12, 2, "",
    "Expenditure", "Nullable")

arcpy.AddField_management(cdaFilePath, "Demand", "DOUBLE", 12, 2, "",
    "Demand", "NULLABLE")

#join two tables

#AddJoin_management (in_layer_or_view, in_field, join_table, join_field,
    {join_type})

joinTable = env.workspace + "\\demog"

arcpy.MakeTableView_management(cdaFilePath, "cdaView")

arcpy.AddJoin_management("cdaView", "DAUID", joinTable, "DAUID")



#CalculateField_management (in_table, field, expression, {expression_type},
    {code_block})

demandCalculation = "((!demog.F1989! + !demog.F1990!
    + !demog.F1991!)*901.2)+(( !demog.F1992! + !demog.F1993!)*1409.2)
    +(( !demog.F1994!+ !demog.F1995!+ !demog.F1996!)
    *1916.4)+(( !demog.F1997!
    +!demog.F1998!)*2490.4)+(( !demog.F1999!)*3678.4)"
```

```
arcpy.CalculateField_management("cdaView", "Demand", demandCalculation,
    expression_type="PYTHON_9.3", code_block="")

arcpy.RemoveJoin_management("cdaView")          #RemoveJoin_management
    (in_layer_or_view, {join_name})

print "3.5 Estimating Demand: "+ str(time.time()-startTime35)+" seconds"



'''

cdaFilePath = env.workspace + "\\RMoWFD\CDA"

arcpy.AddField_management(cdaFilePath, "Expenditure", "DOUBLE", 12, 2, "",
    "Expenditure", "Nullable")

arcpy.AddField_management(cdaFilePath, "Demand", "DOUBLE", 12, 2, "",
    "Demand", "NULLABLE")

joinTable = env.workspace + "\\demog"

arcpy.MakeTableView_management(cdaFilePath, "cdaView")

arcpy.AddJoin_management("cdaView", "DAUID", joinTable, "DAUID")

arcpy.CalculateField_management("cdaView", "Expenditure", '[demog.f2000] * 0.10')

arcpy.CalculateField_management("cdaView", "Demand", '[demog.F1] *
    [CDA.Expenditure]')

arcpy.RemoveJoin_management("cdaView")
```

```
'''

#3.6 creating demand points

#FeatureToPoint_management (in_features, out_feature_class, {point_location})

startTime36=time.time()

demandPoints = env.workspace + "\\demandPoints"

arcpy.FeatureToPoint_management(cdaFilePath, demandPoints, "INSIDE")

print "3.6 Creating Demand Points: "+ str(time.time()-startTime36)+" seconds"

#3.7 Running Location-Allocation in Python

arcpy.CheckOutExtension("Network") # check the extension license

#the sample code for running location-allocation is found at Tool Help in Network
    Analyst Tools -->Analysis --> Make Location-Allocation Layer

#Scenario 1
```

startTime37S1=time.time()

#Create a new location-allocation(L-A) layer. We consider demand travels to facility.

#The parameters below used by MakeLocationAllocationLayer() are defined at the top of this script

#network， impedanceAttribute， Oneway_restriction, and outLayerFile_S1

#MakeLocationAllocationLayer_na (in_network_dataset, out_network_analysis_layer, impedance_attribute,

#

{loc_alloc_from_to}, {loc_alloc_problem_type},

#

{number_facilities_to_find}, {impedance_cutoff}, {impedance_transformation},

#

{impedance_parameter}, {target_market_share},

#

{accumulate_attribute_name}, {UTurn_policy},

#

{restriction_attribute_name}, {hierarchy},

```
#
        {output_path_shape}, {default_capacity}, {time_of_day})


outNALayer = arcpy.na.MakeLocationAllocationLayer (network, "Scenario1",
    impedanceAttribute,
        "DEMAND_TO_FACILITY", "MAXIMIZE_Market_Share",
        12, 17, "LINEAR", #scenario 1; impedence cutoff 17 (min)
                                        "", "",
        ", "ALLOW_UTURNS",
        [Oneway_restriction], "NO_HIERARCHY",
        "STRAIGHT_LINES", "", "")
outNALayer = outNALayer.getOutput(0) #enable the L-A layer to be referenced using
    the layer object.
subLayerNames = arcpy.na.GetNAClassNames(outNALayer)  #Get the names of all
    the sublayers within the L-A layer.


#Store the layer names that we will use later
facilitiesLayerName = subLayerNames["Facilities"]
```

demandPointsLayerName = subLayerNames["DemandPoints"]

#AddLocations(in_network_analysis_layer, sub_layer, in_table, field_mappings, search_tolerance, {sort_field}, {search_criteria}, {match_type}, {append}, {snap_to_position_along_network}, {snap_offset}, {exclude_restricted_elements}, {search_query})

#NAClassFieldMappings(network_analyst_layer, sub_layer_name, {use_location_fields}, {list_candidate_fields})

#Load the existing store locations as the required facility. Set the facility type as Competitor

fieldMappings = arcpy.na.NAClassFieldMappings(outNALayer, facilitiesLayerName)

fieldMappings["Name"].mappedFieldName = "Name"

fieldMappings["Weight"].mappedFieldName = "Demand"

fieldMappings["FacilityType"].defaultValue = 1 #0 candidate, 1 required, 2 competitor

arcpy.na.AddLocations(outNALayer, facilitiesLayerName,origStores,

fieldMappings, "", append = "APPEND",

exclude_restricted_elements = "EXCLUDE")

```
#Load the tract centroids as DEMAND POINTS using default search tolerance. Use
    the field mappings to map the Weight property from Demand field.

demandFieldMappings = arcpy.na.NAClassFieldMappings(outNALayer,
    demandPointsLayerName)

demandFieldMappings["Weight"].mappedFieldName = "Demand"

demandFieldMappings["Name"].mappedFieldName = "Name"

arcpy.na.AddLocations(outNALayer,demandPointsLayerName ,demandPoints,

                demandFieldMappings, "",

                exclude_restricted_elements = "EXCLUDE")


#Solve the location-allocation layer

arcpy.na.Solve(outNALayer)

#Save the solved location-allocation layer as a layer file on disk with relative paths

#SaveToLayerFile_management (in_layer, out_layer, {is_relative_path}, {version})

arcpy.management.SaveToLayerFile(outNALayer,outLayerFile_S1,"RELATIVE")

print "3.7 Location-Allocation Scenario 1: "+ str(time.time()-startTime37S1)+"
    seconds"
```

#Scenario 2

startTime37S2=time.time()

#Create a new location-allocation(L-A) layer. We consider demand travels to facility.

#The parameters below used by MakeLocationAllocationLayer() are defined at the top of this script

#network， impedanceAttribute， Oneway_restriction, and outLayerFile_S2

outNALayer = arcpy.na.MakeLocationAllocationLayer (network, "Scenario2", impedanceAttribute,

"DEMAND_TO_FACILITY", "MAXIMIZE_Market_Share",

1, 17, "LINEAR", #scenario 2: choose 1 optimal locations; impedence cutoff 17 (min)

"", "",

", "ALLOW_UTURNS",

[Oneway_restriction], "NO_HIERARCHY",

"STRAIGHT_LINES", "", "")

outNALayer = outNALayer.getOutput(0) #enable the L-A layer to be referenced using the layer object.

subLayerNames = arcpy.na.GetNAClassNames(outNALayer) #Get the names of all the sublayers within the L-A layer.

#Store the layer names that we will use later

facilitiesLayerName = subLayerNames["Facilities"]

demandPointsLayerName = subLayerNames["DemandPoints"]

#AddLocations(in_network_analysis_layer, sub_layer, in_table, field_mappings, search_tolerance, {sort_field}, {search_criteria}, {match_type}, {append}, {snap_to_position_along_network}, {snap_offset}, {exclude_restricted_elements}, {search_query})

#NAClassFieldMappings(network_analyst_layer, sub_layer_name, {use_location_fields}, {list_candidate_fields})

#Load the candidate store locations as facilities using default search tolerance and field mappings.

fieldMappings = arcpy.na.NAClassFieldMappings(outNALayer, facilitiesLayerName)

fieldMappings["Name"].mappedFieldName = "Name"

```
fieldMappings["Weight"].mappedFieldName = "Demand"

fieldMappings["FacilityType"].defaultValue = 0 # 0 candidate, 1 required, 2
        competitor

arcpy.na.AddLocations(outNALayer, facilitiesLayerName,candidateStores,

                      fieldMappings, "", append = "APPEND",

                      exclude_restricted_elements = "EXCLUDE")



#Load the existing store locations as the required facility. Set the facility type as
        Competitor

fieldMappings = arcpy.na.NAClassFieldMappings(outNALayer, facilitiesLayerName)

fieldMappings["FacilityType"].defaultValue = 2

fieldMappings["Name"].mappedFieldName = "Name"

fieldMappings["Weight"].mappedFieldName = "Demand"

arcpy.na.AddLocations(outNALayer, facilitiesLayerName,origStores,

                      fieldMappings, "", append = "APPEND",

                      exclude_restricted_elements = "EXCLUDE")
```

#Load the tract centroids as DEMAND POINTS using default search tolerance. Use the field mappings to map the Weight property from Demand field.

```
demandFieldMappings = arcpy.na.NAClassFieldMappings(outNALayer,
    demandPointsLayerName)

demandFieldMappings["Weight"].mappedFieldName = "Demand"

demandFieldMappings["Name"].mappedFieldName = "Name"

arcpy.na.AddLocations(outNALayer,demandPointsLayerName ,demandPoints,

                demandFieldMappings, "",

                exclude_restricted_elements = "EXCLUDE")


#Solve the location-allocation layer

arcpy.na.Solve(outNALayer)

#Save the solved location-allocation layer as a layer file on disk with relative paths

#SaveToLayerFile_management (in_layer, out_layer, {is_relative_path}, {version})

arcpy.management.SaveToLayerFile(outNALayer,outLayerFile_S2,"RELATIVE")

print "3.7 Location-Allocation Scenario 2: "+ str(time.time()-startTime37S2)+"
    seconds"
```

```
#Scenario 3

startTime37S3=time.time()

#Create a new location-allocation(L-A) layer. We consider demand travels to facility.

#The parameters below used by MakeLocationAllocationLayer() are defined at the top
        of this script

        #network， impedanceAttribute， Oneway_restriction, and outLayerFile_S3

outNALayer = arcpy.na.MakeLocationAllocationLayer (network, "Scenario3",
        impedanceAttribute,

                "DEMAND_TO_FACILITY", "MAXIMIZE_Market_Share",

                2, 17, "LINEAR", #scenario 3: choose 2 optimal locations;
        impedence cutoff 17 (min)

                                        "", "",

                ", "ALLOW_UTURNS",

                [Oneway_restriction], "NO_HIERARCHY",

                "STRAIGHT_LINES", "", "")
```

outNALayer = outNALayer.getOutput(0) #enable the L-A layer to be referenced using the layer object.

subLayerNames = arcpy.na.GetNAClassNames(outNALayer) #Get the names of all the sublayers within the L-A layer.

#Store the layer names that we will use later

facilitiesLayerName = subLayerNames["Facilities"]

demandPointsLayerName = subLayerNames["DemandPoints"]

#AddLocations(in_network_analysis_layer, sub_layer, in_table, field_mappings, search_tolerance, {sort_field}, {search_criteria}, {match_type}, {append}, {snap_to_position_along_network}, {snap_offset}, {exclude_restricted_elements}, {search_query})

#NAClassFieldMappings(network_analyst_layer, sub_layer_name, {use_location_fields}, {list_candidate_fields})

#Load the candidate store locations as facilities using default search tolerance and field mappings.

fieldMappings = arcpy.na.NAClassFieldMappings(outNALayer, facilitiesLayerName)

```
fieldMappings["Name"].defaultValue = 0

fieldMappings["Name"].mappedFieldName = "Name"

fieldMappings["Weight"].mappedFieldName = "Demand"

fieldMappings["FacilityType"].defaultValue = 0 # 0 candidate, 1 required, 2
    competitor

arcpy.na.AddLocations(outNALayer, facilitiesLayerName,candidateStores,

            fieldMappings, "", append = "APPEND",

            exclude_restricted_elements = "EXCLUDE")


#Load the existing store locations as the required facility. Set the facility type as
    Competitor

fieldMappings = arcpy.na.NAClassFieldMappings(outNALayer, facilitiesLayerName)

fieldMappings["FacilityType"].defaultValue = 2

fieldMappings["Name"].mappedFieldName = "Name"

fieldMappings["Weight"].mappedFieldName = "Demand"

arcpy.na.AddLocations(outNALayer, facilitiesLayerName,origStores,

            fieldMappings, "", append = "APPEND",

            exclude_restricted_elements = "EXCLUDE")
```

#Load the tract centroids as DEMAND POINTS using default search tolerance. Use the field mappings to map the Weight property from Demand field.

demandFieldMappings = arcpy.na.NAClassFieldMappings(outNALayer, demandPointsLayerName)

demandFieldMappings["Weight"].mappedFieldName = "Demand"

demandFieldMappings["Name"].mappedFieldName = "Name"

arcpy.na.AddLocations(outNALayer,demandPointsLayerName ,demandPoints,

demandFieldMappings, "",

exclude_restricted_elements = "EXCLUDE")

#Solve the location-allocation layer

arcpy.na.Solve(outNALayer)

#Save the solved location-allocation layer as a layer file on disk with relative paths

#SaveToLayerFile_management (in_layer, out_layer, {is_relative_path}, {version})

arcpy.management.SaveToLayerFile(outNALayer,outLayerFile_S3,"RELATIVE")

print "3.7 Location-Allocation Scenario 3: "+ str(time.time()-startTime37S3)+" seconds"