School Supply Store Site-Suitability Analysis

Young Jun Jeong | Hasan Nasir | Bhagatinder Longia

Introduction

The aim of this report is to present potential locations for constructing a school supplies store (e.g. Staples, Dollarama) within the city of Mississauga, as well as highlight the steps taken to achieve the results. We found that site-suitability analysis was an appropriate technique to find the potential locations due to its ability to consider various spatial factors and filter accordingly. To maximize efficiency, we will be using tools such as QGIS, PostgreSQL, PostGIS, and open data to aid us in our querying. Throughout the writing of the report, we also took into consideration and further learned about the decisions firms have to make when planning for new store locations. This involves considering strategies for maximizing revenue and customer reach, as well as minimizing competition and potential damage.

In order to better understand the structure of the final query, our research questions and their respective responses are as follows:

What are the components to consider when specifying sites for retail store construction?

- Current land availability/vacancy.
- Proximity to high-demand areas, tailored to specific populations or nearby amenities.
- Acceptable travel distance relative to any potential target customers.
- Adequate distancing from existing retail stores with similar categories of products.
- The ranking in the size out of all available construction sites.
- Extra: Address retrieval of the construction sites to be selected (via Geocoding)

Which criteria should be used to define "suitable" sites for construction?

- Subcategories of city land use
- Wildcard keyword for filtering existing similar retail stores out
- Distance for proximity to the nearby amenities in high demand
- Buffer distance from the existing retail stores in similar category
- Address name selection for Geocoding

Data Collection

The datasets used in this study are:

1. Land Use from Mississauga City Open Data Portal (Polygon Shapefile)
2. Business Directories from Mississauga City Open Data Portal (Point Shapefile)
3. Watercourse → Peel Region Open Data Portal (Polyline Shapefile)
4. Digital Elevation Model → Peel Region Open Data Portal (Raster File)
5. Address Points from Mississauga City Open Data Portal (Point Shapefile)
6. Google Map (Reference Web Map)

1.

The new store has to be built on vacant land, so the first step was to find all instances of vacant lands from the land use dataset. In addition to land vacancy, it would be ideal for the store to be built near a school, as the target customers will primarily be students. To solve both these problems, we obtained a polygon shapefile of Mississauga Land Use data from the Mississauga City Open Data Portal. The important attributes used from this dataset are listed below:

Attribute	Data Type	Description
gid (created by us using the shp2pgsql command)	Spatial Index (Integer)	Unique identifier for all the land polygons within the dataset
geom (created by us using the shp2pgsql command)	Geometry	Stores geometric information for each polygon that represents a land.
landuse	String	Indicates to us what each land in the dataset is being used for.

The dataset originally had a WGS 1984 (EPSG: 4326) coordinate system, but we transformed the coordinate system to NAD 1983 UTM Zone 17N (EPSG: 26917) while loading this shapefile into our database:

```
-- Loading Landuse dataset:
shp2pgsql -D -s 4326:26917 -g geom -I 2021_Existing_Land_Use.shp landuse.landuses > landuse.sql
psql -h localhost -U postgres -p 5432 -d ggr381_gp -f landuse.sql
```

2.

While making sure the store was located as close as possible to schools, we also wanted our store to be distant from any competitor stores that also sold office and school supplies. This was to maximize visits to our store and provide a healthy gap between competitors to maintain a steady business. To do this, we obtained a Point shapefile of all the Mississauga business directories from the Mississauga City Open Data Portal. The important attributes used from the dataset are listed below:

Attribute	Data Type	Description
gid	Spatial Index (Integer)	Unique identifier for all the geometries (points)
name	String	Name of the business
naicstitle	String	Category of the business
naicsdescr	String	Detailed description of the business
geom	Geometry	Stores geometric information of each point that represents a business directory

The dataset originally had a WGS 1984 (EPSG: 4326) coordinate system, but we transformed the coordinate system to NAD 1983 UTM Zone 17N (EPSG: 26917) while loading this shapefile into our database:

```
-- Loading Business Directory dataset:
shp2pgsql -D -s 4326:26917 -g geom -I 2022_Mississauga_Business_Directory.shp business.buisnesses > business.sql
psql -h localhost -U postgres -p 5432 -d ggr381_gp -f business.sql
```

3.

For safety reasons, we wanted to enforce our store's location to be at least 300 meters away from any watercourse, to ensure ground stability and lessen any impact on the surrounding land from water-related disasters. To do this, we obtained watercourses polyline data that contains the rivers and streams segments existing over the Peel Region.

Attribute	Data Type	Description
gid	Spatial Index (Integer)	Unique identifier for all the geometries (polylines)
mun_name	String	The name of the municipality through which the watercourse passes
geom	Geometry	Stores geometric information of each polyline, representing a river/stream segment in the Peel Region.

The dataset originally had a WGS 84 / Pseudo-Mercator -- Spherical Mercator (EPSG: 3857 coordinate system, but we transformed the coordinate system to NAD 1983 UTM Zone 17N (EPSG: 26917) while loading this shapefile into our database:

```
-- Loading Watercourses Dataset:
shp2pgsql -D -s 3857:26917 -g geom -I Watercourse.shp watercourse.watercourses > watercourses.sql
psql -h localhost -U postgres -p 5432 -d ggr381_gp -f watercourses.sql
```

4.

To minimize construction costs for our new store, we wanted any final locations to have an average slope/incline value less than or equal to 6 degrees. In order to do so, we obtained a DEM (Digital Elevation Model) of Peel Region from the Peel Region Open Data Portal. We preprocessed this DEM in QGIS by making a slope raster from it. We then used Clip Raster on the slope raster using the polygon layer of all the vacant lands in Mississauga (derived from the land use dataset) as the mask. This gave us the slope value raster for only the cells that were within vacant lands of Mississauga.

The raster originally had a WGS 1984 (EPSG: 4326) coordinate system, but we transformed the coordinate system to NAD 1983 UTM ZONE 17N (EPSG: 4326) in QGIS before loading this shapefile into our database:

```
-- Loading raster
raster2pgsql -s 26917 -I -C -M -F -t auto Slope_Vacant.tif slope.slopes > slope.sql
psql -h localhost -p 5432 -U postgres -d ggr381_gp -f slope.sql
```

Once the raster was loaded into our database, it had the following attributes:

Attribute	Data Type	Description
rid	Integer	Unique identifier for all the raster cells in the dataset.
rast	Geometry	Stores the information that is contained in each raster cell.

5.
We also wanted the suitable vacant lands to contain address information. For this, we obtained a Point shapefile containing the addresses from the Mississauga City Open Data Portal. The important attributes that we used from this dataset are listed below:

Attribute	Data Type	Description
gid	Spatial Index (Integer)	Unique identifier for all the geometries (points) which represent addresses
streetnum	String	The numbering of the street
streetname	String	The name of the street
streettype	String	The type-abbreviation of the street
geom	Geometry	Stores geometric information of the points which represent addresses in Mississauga

The dataset originally had a WGS 84 / Pseudo-Mercator -- Spherical Mercator (EPSG: 3857) coordinate system, but we transformed the coordinate system to NAD 1983 UTM Zone 17N (EPSG: 26917) while loading this shapefile into our database:

```
-- Loading Address Points Dataset:
shp2pgsql -D -s 3857:26917 -g geom -I AddressPoints.shp address.addresses > address.sql
psql -h localhost -U postgres -p 5432 -d ggr381_gp -f address.sql
```

Data Analysis (SQL Querying)

1.
To ensure our store's location be as far as possible from competitor supply stores, we used the business directory linked above to find all competitor stores in Mississauga.

We did this with the following query:

```
-- 1.1 Creating Temporary Table of Business Directories to Avoid (Competitors)
CREATE TEMPORARY TABLE avoid AS
    SELECT *
    FROM business.businesses
    WHERE naicsdescr ILIKE '%general merchandise%'
    OR naicsdescr ILIKE '%Office Supplies%';
```

When observing the business directory dataset, we noticed that the stores we wanted to avoid contained keywords such as "Office Supplies" or "general merchandise" in their dataset description, so we created a temporary table called **avoid** to specifically store those businesses. The ILIKE command was used to find the records (stores) in the business directory table that contained either the 'general merchandise' or 'Office Supplies' keywords in their description.
We also wanted to keep a radius around our competitor stores to ensure a minimum distance from our store, therefore we chose to make a 1 km buffer around each of the stores in the temporary avoid table.

```
-- 1.2 1km Buffering Avoidable Business Locations
CREATE TEMPORARY TABLE temp_buffered AS
    SELECT gid,
           "name",
           ST_Buffer(geom, 1000) AS buffered_geom
    FROM avoid;
```

The ST_BUFFER function created these new buffers and stored the geometry, names, and gid of the stores that these buffers surrounded in a new temporary table called **temp_buffered**.

Using the ST_UNION function, we then made a unified buffer that combined every buffer stored in the **temp_buffered** in another table called **unified_buffer** as follows:

```
-- 1.3 Unify Buffers into One
CREATE TEMPORARY TABLE unified_buffer AS
    SELECT ST_Union(buffered_geom) AS unified_geom
    FROM temp_buffered;
```

This unified buffer thus represents the total area in which we cannot build our store (due to proximity from competitor stores).

2.

The next step was to find vacant lands within 500 meters from schools, as we wished for our store to be frequented by students.

We did this by using the following query on the land use dataset:

```
-- 2. Unique Vacant Land that are within 500m from School
CREATE TEMPORARY TABLE near_school AS
    SELECT DISTINCT l1.gid AS gid,
    l1.landuse AS landuse,
    l1.geom AS geom
    FROM landuse.landuses l1
    INNER JOIN landuse.landuses l2
    ON ST_DWithin(l1.geom, l2.geom, 500)
    WHERE l1.landuse = 'Vacant' AND l2.landuse = 'School';
```

This query identifies the information of the vacant land polygons that are located within a 500-meter radius from school properties by doing a self join on the **land use** table using the ST_DWithin spatial function. We do a self join on the table because both of the land use types (vacant lands and school lands) exist in the same dataset. The ST_Dwithin function helps us to find the vacant land polygons which were within 500 meters of the school polygons. Some vacant land polygons were within 500 meters of multiple school polygons, and therefore were appearing more than once. To combat this we used the DISTINCT keyword to ensure that each vacant land appeared only once in the table. The result of this query was stored in a table called **near_school**.

3.

After finding the vacant lands located near schools, we further filtered them by spatially analyzing whether they lied within the unified buffer (representing the area that we cannot build in due to the presence of competing stores). We also wanted to compute the area of such lands in square-meters and acres. Therefore we used the following query and stored the results in the temporary table named **out_buffer**.

```
-- 3. Vacants NOT Within Unified Buffer + Adding Area Size Column
CREATE TEMPORARY TABLE out_buffer AS
    SELECT *,
            ST_Area(near_school.geom) AS AREA_SQM,
            ST_AREA(near_school.geom)/4046.85642 AS AREA_ACRE
    FROM near_school
    LEFT JOIN unified_buffer ub
    ON ST_Within(near_school.geom, ub.unified_geom)
    WHERE ub.unified_geom IS NULL;
```

This query identifies the vacant lands that are not within the unified buffer and adds two columns to the result: one for the area in square meters (AREA_SQM) and another for the area in acres (AREA_ACRE). The ST_Area function was used to compute the area of the geometries in square meters since meters are the default calculation unit of the NAD 1983 coordinate system. This computed area was by 4046.85642 to find the equivalent area in acres.

4.

After having filtered the vacant lands based on proximity to schools and distance from competitor stores, we also wanted to make sure our store did not lie within 300 meters of any watercourses in Mississauga due to the potential risk from floods.

In order to do this, we first made a 300 meter buffer around all the watercourses in Mississauga using the following query:

```
-- 4. Distancing from Watercourses for Potential Flood Extent

-- 4-1. 300m Buffering from Watercourses
CREATE TEMPORARY TABLE water_buffer_indiv AS
    SELECT gid,
           "type",
           ST_Buffer(geom, 300) AS buffered_geom
    FROM watercourse.watercourses
    WHERE mun_name = 'Mississauga';
```

The table **water_buffer_indiv** is created, containing the gid, type of the watercourse, and the buffer geometry of each individual geometry of itself, with the distance parameter of 300 meters, under the condition where the name of the municipality is only Mississauga.

We then unified all the buffers into one by using the following query:

```
-- 4-2. Unify Buffers into One
CREATE TEMPORARY TABLE unified_water_buffer AS
    SELECT ST_Union(buffered_geom) AS unified_water_geom
    FROM water_buffer_indiv;
```

The temporary table **unified_water_buffer** is created, containing only the unified polygon geometry, composed of the buffered geometry column from **water_buffer_indiv**, into a field: unified_water_geom, retrieved from the temporary table previously created: **water_buffer_indiv**.

Finally, we filtered out the vacant lands that were not within 300 meters of any water courses (did not lie in the unified buffer) using the following query:

```sql
-- 4-3. Sort Out Filetered Vacant Land from Unified Water Buffer
CREATE TEMPORARY TABLE flood_safe AS
    SELECT *
    FROM out_buffer ob
    LEFT JOIN unified_water_buffer uw
    ON ST_Within(ob.geom, uw.unified_water_geom)
    WHERE uw.unified_water_geom IS NULL;
```

The temporary table **flood_safe** is created, retrieving all the columns from the left join of the table: **out_buffer** and the table: **unified_watter_buffer**, on a spatial joining key under the condition where the geometry of **out_buffer** is within the geometry of **unified_water_geom**.

5.

After filtering the vacant lands based on proximity from water courses, we had to filter them again according to the slope of the ground (to minimize the cost of construction). To this, we first found the value of the slope (in degrees) of the slope raster cells that were within our filtered vacant lands using the following query:

```sql
-- 5. Slope Limiting
-- 5.1 Find slope value of each cell that is within the filtered vacant lands
CREATE TEMPORARY TABLE temp_summary AS
    SELECT fls.gid AS gid,
            (ST_SummaryStats(s.rast)).mean AS mean,
            fls.geom AS geom,
            fls.area_sqm AS area_sqm,
            fls.area_acre AS area_acre
    FROM flood_safe AS fls
    INNER JOIN slope.slopes AS s
    ON ST_Intersects(fls.geom, s.rast);
```

The temporary table **temp_summary** is created from the inner join of the table: **flood_safe** and **slopes** table stored in **slope** schema, under the spatial join condition where the geometry of the **flood_safe** intersects with the **slope**'s raster cell values 'rast' in grid containing the **flood_safe**'s gid, geometry, area in square metres and acres calculated, with a new field of calculated values of the summary statistics of the raster but retrieving the mean subcolumn, as into slope mean value.

We then computed the average slope of each filtered vacant lands by taking the average of the slope values of each cell that fell within that filtered vacant land using the following query:

```sql
-- 5.2 Find average slope of the cells that are within each fitered vacant land
CREATE TEMPORARY TABLE slope_calculated AS
    SELECT gid,
            AVG(mean) AS average
    FROM temp_summary
    GROUP BY gid;
```

The temporary table **slope_calculated** is created from the **temp_summary,** aggregating the multiple mean slope values in the field per each geometry, due to the partitioned raster calculation aligned to the different raster cell size during the preprocessing

Finally, we created a new table **slope_satisfied** that contained only the information about the lands and their average slope for lands which had an average slope of less than or equal to 6 degrees. This was done by using the following query:

```sql
-- 5.3 Join Tables to Insert Average Slope as a column for each filtered land
-- and show only lands that have an average slope of less than 6 degrees
CREATE TEMPORARY TABLE slope_satisfied AS
SELECT DISTINCT ts.gid,
        ts.area_sqm,
        ts.area_acre,
        sc.average AS average_slope,
        ts.geom
FROM temp_summary As ts
RIGHT JOIN (SELECT *
            FROM slope_calculated
            WHERE average <= 6) As sc
ON ts.gid = sc.gid;
```

The temporary table **slope_satisfied** contains the **temp_summary**'s geometric ID, area calculated in square metres and acre units, the average slope value in a double format, with the geometry of **temp_summary**. It is right-joined from the **temp_summary** table to the subquery result table (a subset of **slope_calculated** where it satisfies the average slope value is less than or equal to 6) by the join-key matching with the geometric object ID.

6.

Now that we had our final results for acceptable vacant lands, we added address information to them via geocoding by using the address points dataset that we obtained by using the following queries:

The **geocode** temporary table stores all the necessary attributes needed for the geocoding of each filtered suitable site candidate.

```sql
--- 6.1 Address Geocoding by JOIN
CREATE TEMPORARY TABLE geocoded AS
    SELECT DISTINCT ss.gid AS gid,
            ss.area_sqm AS AREA_SQM,
            ss.area_acre AS AREA_ACRE,
            ad.streetnum || ' ' || ad.streetname || ' ' || streettype AS Full_Address,
            ss.geom AS vacant_land_geom,
            ss.geom as address_points_geom
    FROM slope_satisfied ss
    LEFT JOIN address.addresses ad
    ON ST_Within(ad.geom, ss.geom)
    WHERE ad.municipali = 'Mississauga' AND ad.streettype IS NOT NULL;
```

The **geocoded** temporary table is created from the left-join of **slope_satisfied** table to **addresses** table in **address** schema, under the spatial join condition under when the geometry of **addresses** table is within the geometry of **slope_satisfied**, filtered only where the **addresses** table's municipality field value is Mississauga and streettype field value is not a null. Then, the distinct records of the **slope_satisfied**'s geometric ID, area size calculated in square metres and acre units, geometries of suitable sites and geocoded address points, and **addresses**'s concatenated address elements unified, are stored.

The **geocoded_aggregate** temporary table stores the all street information as an array of full address names as each candidate site may contain one or multiple locations geocoded.

```
-- 6.2 Aggregate Addresses as a List
CREATE TEMPORARY TABLE geocoded_aggregate AS
  SELECT
    gid,
    array_to_string(array_agg(Full_Address), ', ') AS Full_Address_List
  FROM geocoded
  GROUP BY gid;
```

The temporary table **geocoded_aggregate** stores the array form of the collection of all possible geocoded locations along with the respective geometric ID, aggregated using the grouping of the records by the geometric ID.

The **suitable_sites** now finally stores all the information from the sites satisfying all the aforementioned conditions and restrictions, which allows the audiences to check the data table starting from the largest sized-sites as a default setting.

```
-- 6.3 Creating Table as a Final Result
CREATE TABLE suitable_sites AS
  SELECT DISTINCT gc.gid AS gid,
    gc.AREA_SQM AS AREA_SQM,
    gc.AREA_ACRE AS AREA_ACRE,
    ga.Full_Address_List AS Full_Address_List,
    gc.vacant_land_geom AS vacant_land_geom
  FROM geocoded AS gc
  INNER JOIN geocoded_aggregate AS ga
  ON gc.gid = ga.gid
  ORDER BY gc.AREA_ACRE DESC;
```

The temporary table **suitable_sites** is created as an inner-join of the **geocoded** table and the **geocoded_aggregate** table by matching join-key: the geometric ID, and it stores **geocoded** table's geometric ID, area size calculated in square metres and acres unit, the an array list of the full address names, and the geometry of each unique potential candidate sites available. Then, ordered by the area size in acre units of the suitable sites, starting from the site with the largest area calculated.

Data Visualization

To visualize our final results, we used the following query to select the attributes we wanted to show on the final table:

```
-- 7. Retreive Table Results
SELECT gid AS GID,
        AREA_SQM AS "Area: Square Metres",
        AREA_ACRE AS "Area: Acres",
        Full_Address_List AS "Full Address List",
        vacant_land_geom AS "Vacant Land Geometry"
FROM suitable_sites
ORDER BY AREA_ACRE DESC;
```

The Appendix contains a comprehensive screenshot of the table listing the final results, with the following attributes:

- unique geometric object id
- area size (square meters & acres unit)
- list of geocoded addresses
- the geometry of the constructable sites in polygon

The map below visualizes the final locations of the vacant lands we identified as satisfying all of our site-suitability criteria, on a map of Mississauga, via QGIS.

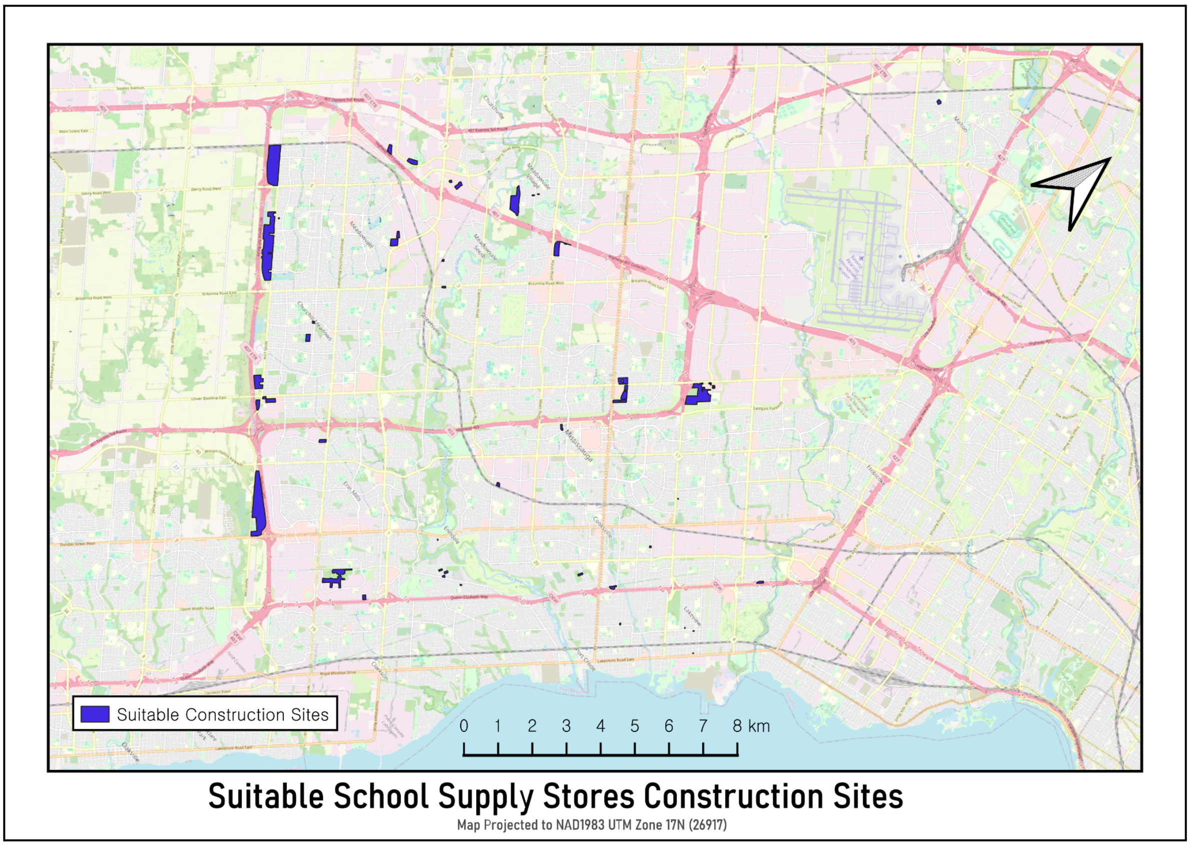

Fig 1. Map of Suitable Supply Store Sites

Results

In summary, our site suitability analysis successfully filtered and identified vacant land that conforms to our criteria using a combination of PostgreSQL, the PostGIS spatial extension, and QGIS for the followed-up data visualization as a static map.

Initially we had 503 vacant lands to choose from, and after ensuring that our vacant lands satisfied the following criteria:

- the land is farther than 1000 meters away from rival businesses
- the land is within 500 meters from schools
- the land is farther than 300 meters from watercourses
- the average slope of the land is less than 6.0 degrees-rise

We narrowed down our choices to 50 vacant lands, all of which maximize safety and accessibility to potential customers while minimizing any potential hazards.

An ideal vacant land would satisfy all of the above requirements, and possibly more, with some possible filtering and analysis criteria below:

- Accounting for population density, aiding us in identifying where shop locations may be most beneficial.

- Filtering for land size, ensuring adequate room is available to build a store large enough to satisfy population needs.

- Evaluating based on land price, which contributes to allow better budgeting and resource allocating.

- Considering proximity to highways/arterial roads, vastly improving connectivity to the store and ensuring easy transportation and accessibility.

Some potential improvements for the project include:

- Leveraging spatial data more effectively by reducing the number of tables used.

- Attempting more complex spatial analysis for more accurate results, or to verify validity of outputs.

Appendix

	gid integer	Area: Square Metres double precision	Area: Acres double precision	Full Address List text	Vacant Land Geometry geometry
1	8727	496582.7134265451	122.70826090403897	6136 NINTH LINE, 6168 NINTH LINE, 6252 NINTH LI...	010600002025690000010000000
2	7477	470533.34376157133	116.27132147217897	3005 NINTH LINE, 3115 NINTH LINE, 3415 NINTH LI...	010600002025690000010000000
3	14439	385742.4178214264	95.31902736035947	7010 NINTH LINE, 7020 NINTH LINE, 7044 NINTH LI...	010600002025690000010000000
4	5261	243898.80879626944	60.26870822273192	4496 TOMKEN RD, 4598 TOMKEN RD, 4896 TOMKEN ...	010600002025690000010000000
5	4700	144880.92273462945	35.80085570088732	1149 LAMPLIGHT WAY, 1150 LAMPLIGHT WAY	010600002025690000010000000
6	3111	142866.2213778918	35.303012153293004	2495 SHERIDAN PARK DR, 2505 SHERIDAN PARK DR,...	010600002025690000010000000
7	25425	85266.84316700387	21.069895819779013	4110 RAYBRIA WAY, 4112 RAYBRIA WAY, 4114 RAYB...	010600002025690000010000000
8	22956	61125.54826745012	15.104451930975628	6370 BELGRAVE RD	010600002025690000010000000
9	24903	44593.66755401658	11.019335238490271	125 EGLINTON AVE, 131 EGLINTON AVE, 43 EGLINTO...	010600002025690000010000000
10	5056	42036.251398480024	10.387383943431336	2460 SURVEYOR RD, 2468 SURVEYOR RD, 2478 SURV...	010600002025690000010000000
11	4911	36110.09188026606	8.922998034179344	3539 PLATINUM DR, 3541 PLATINUM DR, 3545 PLAT...	010600002025690000010000000
12	11829	34052.16064935802	8.414472152031037	2305 MEADOWVALE BLVD	010600002025690000010000000
13	21173	32970.54861415209	8.147199997313493	105 ELIA AVE	010600002025690000010000000
14	2021	26264.628842621627	6.490131133098521	4452 NINTH LINE	010600002025690000010000000
15	6839	25387.9087014736	6.273488868051711	2385 MEADOWPINE BLVD	010600002025690000010000000
16	16669	24326.642621023817	6.011244308248479	5450 FESTIVAL DR	010600002025690000010000000
17	10389	21776.72053601141	5.381144838346256	6855 FINANCIAL DR, 6875 FINANCIAL DR	010600002025690000010000000
18	15699	16780.7617293379	4.146616530896814	4200 WINSTON CHURCHILL BLVD, 4202 WINSTON C...	010600002025690000010000000
19	21388	14491.18230585728	3.580849133722733	4615 HURONTARIO ST	010600002025690000010000000
20	22139	13540.58523487539	3.345951481736876	2445 SURVEYOR RD	010600002025690000010000000
21	27391	13084.779308028337	3.2333193842415433	136 EGLINTON AVE	010600002025690000010000000
22	18446	12904.834630331896	3.188854085990008	2020 CAMILLA RD, 2026 CAMILLA RD, 2032 CAMILL...	010600002025690000010000000010
23	26918	12531.549370707917	3.0966132894598513	2251 NORTH SHERIDAN WAY, 2253 NORTH SHERIDA...	010600002025690000010000000010
24	14553	11393.430166233054	2.815377909116196	110 ELIA AVE	010600002025690000010000000010
25	17272	10693.710337336699	2.642473373774081	3233 BRANDON GATE DR	010600002025690000010000000010
26	25713	9896.556668667681	2.445492412272853	1560 BRENTANO BLVD, 2004 LAUGHTON AVE	010600002025690000010000000010
27	15634	8551.001690522093	2.1129985359159575	1004 CENTRAL PKY, 1006 CENTRAL PKY, 1008 CENT...	010600002025690000010000000010
28	24496	8354.015167233687	2.0643221059060175	353 RATHBURN RD, 355 RATHBURN RD, 357 RATHBU...	010600002025690000010000000010
29	19040	7026.914099550602	1.7363882901362244	2170 GORDON DR	010600002025690000010000000010
30	8574	6210.246989299506	1.5345854522062599	2229 DOULTON DR	010600002025690000010000000010
31	10563	4813.109863178361	1.1893453494894095	2 JOSEPH ST	010600002025690000010000000010
32	10909	4703.775275562868	1.1623281844931055	4611 TOMKEN RD, 4625 TOMKEN RD	010600002025690000010000000010
33	5959	4444.043149803633	1.098146978440029	2250 DOULTON DR, 2270 DOULTON DR	010600002025690000010000000010
34	15633	4184.500631349269	1.0340126253723794	6918 FINANCIAL DR	010600002025690000010000000010
35	6699	3596.0323405869167	0.8885989438159797	3580 THOMAS ST, 5590 TENTH LINE	010600002025690000010000000010
36	14162	3581.641235827079	0.8850428243824571	2208 DOULTON DR	010600002025690000010000000010
37	1267	3021.7060622053586	0.7466798296257219	4930 TOMKEN RD	010600002025690000010000000010
38	17017	2873.061350563928	0.709948921430706	11 ALPHA MILLS RD	010600002025690000010000000010
39	19951	1619.351052527442	0.4001503598013596	6842 SECOND LINE	010600002025690000010000000010
40	16963	1543.7108729900867	0.3814592643714517	1338 BROADMOOR AVE, 1346 BROADMOOR AVE	010600002025690000010000000010
41	16378	1398.3113792393526	0.3455302669817360	2110 STONEHOUSE CRES	010600002025690000010000000010
42	23521	1050.8971099765386	0.2596823313974995	6850 CAMPBELL SETTLER CRT	010600002025690000010000000010
43	900	767.9514136448125	0.1897649271294909	6697 METEOR CRT, 6701 METEOR CRT, 6705 METEO...	010600002025690000010000000010
44	15209	747.8166650463024	0.1847895223933599	2412 HENSALL ST	010600002025690000010000000010
45	24053	745.9369819320192	0.1843250425800921	2420 HENSALL ST	010600002025690000010000000010
46	22073	722.5679886304505	0.1785504385723797	1234 MEREDITH AVE	010600002025690000010000000010
47	12821	696.9840753564192	0.1722285159196281	1571 ADAMSON ST	010600002025690000010000000010
48	18764	478.37406570494926	0.1182088060601243	720 HILLMAN CRES	010600002025690000010000000010
49	3257	424.7086257037618	0.1049477870291632	1015 MEREDITH AVE	010600002025690000010000000010
50	453	416.9726989188489	0.1030361978888415	882 SIXTH ST	010600002025690000010000000010

Fig 2. Table of Suitable Supply Store Sites (Ordered by the Largest Land Size)