Spatial clustering for generation of routes adjusted to the vehicle fleet in spatial databases

Alvaro Enrique ORTIZ and Luz Angela ROCHA, Colombia

Key words: Route algorithm, route optimization, spatial databases processing, transportation logistics

SUMMARY

A new clustering algorithm is proposed to adjust to logistical problems of optimal routes in cases where there is an already established fleet of vehicles, joined with a cloud of points that must be visited with the fleet of vehicles available over a road network in a spatial database. The algorithm developed was implemented in a spatial database (PostgreSQL-PostGIS) and determines the groups that each vehicle must go through, taking into account the particular capacity of each vehicle, as well as the order of the places the route should have. It makes use of Dijkstra's algorithm to determine the optimal route between every two consecutive places on the tracks registered in the database. The number of groups generated corresponds to the number of vehicles available, or until the company no longer has more places to visit, whichever comes first.

The algorithm is tested with a practical case on the roads of the city of Bogotá (Colombia) to determine the routes that the buses of a school must travel to pick up and drop off students on their journey from home to school and vice versa. It should be noted that all vehicle information, including student capacity, student information, which includes their residence address, is stored in the spatial database. The database is complemented with information on roads (polylines), home plates (points) and blocks (polygons). The school case demonstrates that the algorithm is functional and is adjustable to the number of groups and quantity of capacity measurement for each vehicle, in addition to the order of travel for each route, which allows us to conclude that it is a very practical algorithm, adjustable and applicable in many environments of logistics solutions with only the use of spatial databases and a geographic information system that serves as a graphic viewer of the routes established on the roads of the spatial work environment.

Spatial clustering for generation of routes adjusted to the vehicle fleet in spatial databases

Alvaro Enrique ORTIZ and Luz Angela ROCHA, Colombia

1. INTRODUCTION

Many of the transport logistics problems require careful planning that should optimize the available resources in order to reduce costs and simultaneously expedite travel times. If you have the spatial information of the environment in which the routes are developed, plus the information of the start and destination of the routes, it is possible to automate an effective planning of the different routes that use the available transport capacity, and manage to optimize the time and resources invested.

Data science (Brodie, 2019) is a multidisciplinary approach that allows data to be used to obtain useful information, which means that it must transform, organize, model, analyze, visualize and communicate the information that is available, in order to achieve get the data you want. In this same sense, spatial data science (Bação et al., 2020), obtains useful information from spatial data in combination with other data.

Route planning is a logistics problem for many companies, although there are routing algorithms, they are insufficient in many cases, requiring custom applications as in the case (Da et al., 2017). If we apply spatial data science to automate the effective planning of the routes that are required to be carried out with the vehicles available or necessary to carry out the work, travel times and therefore operating costs can be reduced. The information of the routes can be visualized graphically on a map in the operating environment.

2. CASE STUDY

As a case study to carry out a practical approach, information is taken from a school that must cover the routes of its students from the house of each one of them to the school headquarters, and of course back to the respective houses of the students. students.

A school is chosen from the network of district schools in the city of Bogotá, D.C. (Colombia), and the information of 250 students in a radius of 6 kilometers around the school will be simulated. Students' contact information is held in the school's database, including the address, which is used to determine how far the route needs to go.

Information regarding the city is required, such as blocks, roads, schools, nomenclature axes, house plates, which will be used in the information analysis process to automate the route selection process.

The database that contains all the spatial and student information is in PostgreSQL and makes use of the PostGIS spatial extension for the management and analysis of spatial information, as well as pgRouting to apply the routing algorithms. The QGIS geographic information system software will also be used as an information viewer.

3. DATA AND SOURCES OF INFORMATION

The primary source of information regarding Bogotá is obtained from the Spatial Data Infrastructure of the Capital District (IDECA). Each school must have a database of its students where the residence addresses of the students are registered, in this example we will use a table created in PostgreSQL where we will register a name and an address for each of the 500 students, we will fill the Student identifier field with a sequential number, the student's name with the text “Student” and the identifier number, and a random address within a radius of between 1000 and 6000 meters around the school, the Student table created in the Database will be filled automatically by some functions in PostgreSQL.

The IDECA information that will be used in the application will be:

- Blocks, 43,887 records
- Track lines, 139,089 records
- Residential Plates, 1,789,583 records
- Schools, 2,539

A school will be randomly chosen, to which the information from the student table will be associated, see figure 1.

The addresses of the students will also be randomly selected within a radius of 6,000 meters around the school, the distance being greater than 1,000 meters, since students who live near the school can walk to it.

For the case study, the student table is also generated automatically, with a sequential identifier between the numbers 1 and 250, and the name will be 'Student' adding the identifier number, Table 1 shows the first records of the student table.

Table 1, First records of the students' table in the database

	id [PK] bigint	nombre character varying (30)	direccion character varying (50)
1	1	Estudiante 1	KR 111D # 67 08
2	2	Estudiante 2	CL 64C # 104 53
3	3	Estudiante 3	CL 75A # 77B 35
4	4	Estudiante 4	KR 100C # 129C 77
5	5	Estudiante 5	CL 126C # 118A 65

Of course, a school will have more attributes associated with students, and it will surely have a table schema that includes subjects, grades, teachers, etc., but this information is irrelevant for the purposes of the case study exercise.

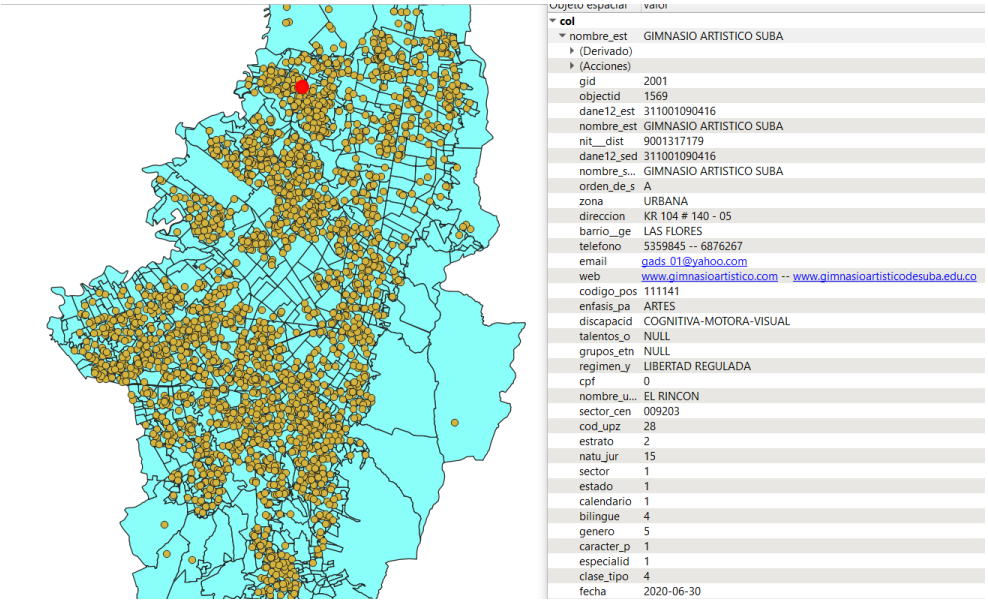

Figure 1 - View of the schools in Bogotá and the selected school.

4. SPATIAL ANALYSIS PROCESS

To carry out the different analyses that are carried out with the spatial information, several algorithms will be used that have to do with geocoding, grouping, determination of the order of travel and optimal routes.

4.1 Geocoding

Initially, it is required to carry out a process of georeferencing the address of each one of the students, this process is called Geocoding (McDonald et al., 2017), and consists of the conversion of descriptive location data, such as a postal address, or a designated place, in an absolute geographic reference (Wilson & Knoblock, 2007). Geocoding is based on a matching algorithm that tries to determine the location of the address in a range of addresses from a reference data set (Owusu et al., 2017).

Usually, the matching algorithm is integrated into an address locator, which creates a geometry for textual descriptions of addresses. Geocoding of streets is the most widely used technique, where the algorithm performs a linear interpolation of the address within a range of address numbers and the polarity of the street segment. Figure 2 describes the general process performed by the geocoding algorithms.

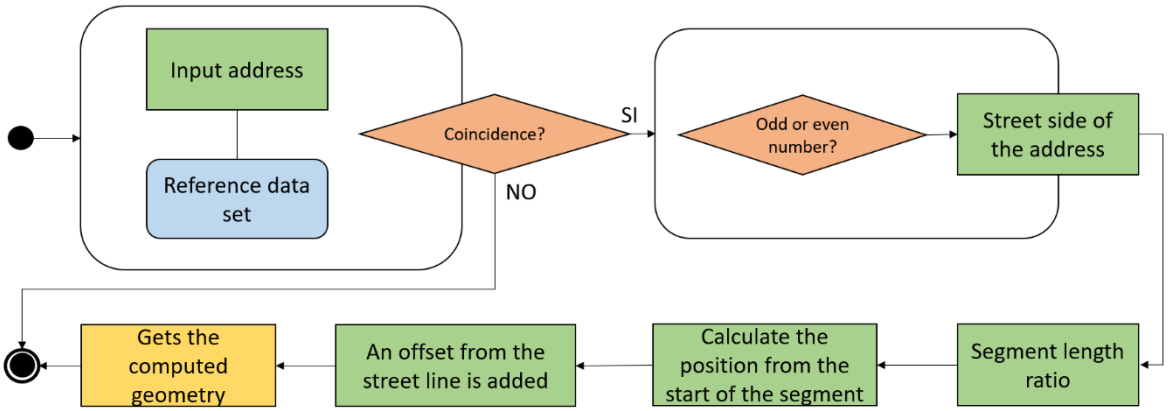

Figure 2 - Geocoding process.

In PostgreSQL there are functions to carry out the geocoding process, generating, from the input address, a set of possible locations that include a point geometry with a qualification, the lower the qualification, the more likely the match will be. In order to use these functions it is necessary to install the postgis, postgis_tiger_geocoder (Honduvilla & Manso Callejo, 2010), fuzzystrmatch and address_standardizer extensions, modifications must be made so that the Bogotá address system works with the geocoder procedures.

Fortunately, IDECA (Spatial Data Infrastructure of Bogotá, s.f.), has a spatial layer that includes the residence plates that include the address as an attribute, so this layer can be used to identify the address, and the student that matches is associated with the address of the database, obtaining the housing locations of the students as shown in figure 3, where the yellow dots correspond to the student housing and the red dot to the school.

It should be noted that geocoding presents errors in the point estimation of the input address, since interpolation is a statistical process that is based on the registered number of the address, which is also an estimate of distance.

Having the georeference of the addresses, now it is necessary to group them by proximity to each other to identify the students of each route. There are two cases that can arise: 1- there is a specific number of buses and the number of students must be accommodated to them, and 2- students are grouped by group size and from the result it is known how many buses are required to cover routes.

Figure 3 - Geocoding of student addresses.

4.2 Clustering – K-means

Clustering is the task of grouping a set of objects so that similar objects end up in the same group and different objects are separated into different groups (Shalev-Shwartz, 2014), for our case, the similarities between the objects correspond to the closeness in Euclidean distance. Perhaps the simplest algorithm to detect groups of data is known by the name of k-means (Morissette & Chartier, 2013), the "k" in the name indicates the requested number of clusters, a parameter whose value is provided by the user. (Kubat, 2017). Figure 4 illustrates the general process of the K_Means algorithm.

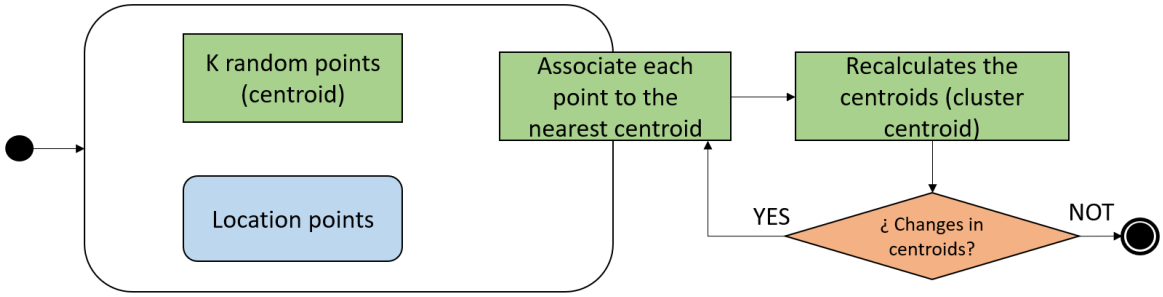

Figure 4 - Clustering process with K-Means

This algorithm adjusts to the case in which there is a specific number of buses (K cluster) to accommodate the route to the students of each group, but the maximum number of cluster members is not controlled.

4.3 Clustering – DBScan

Another widely used clustering algorithm is the DBScan algorithm, it is used for density-based spatial clustering of noisy applications, proposed in (Ester et al., 1996), where it models clusters as clusters of high point density, where if a point belongs to a cluster must be close to a lot of points of said cluster, see process in figure 5.

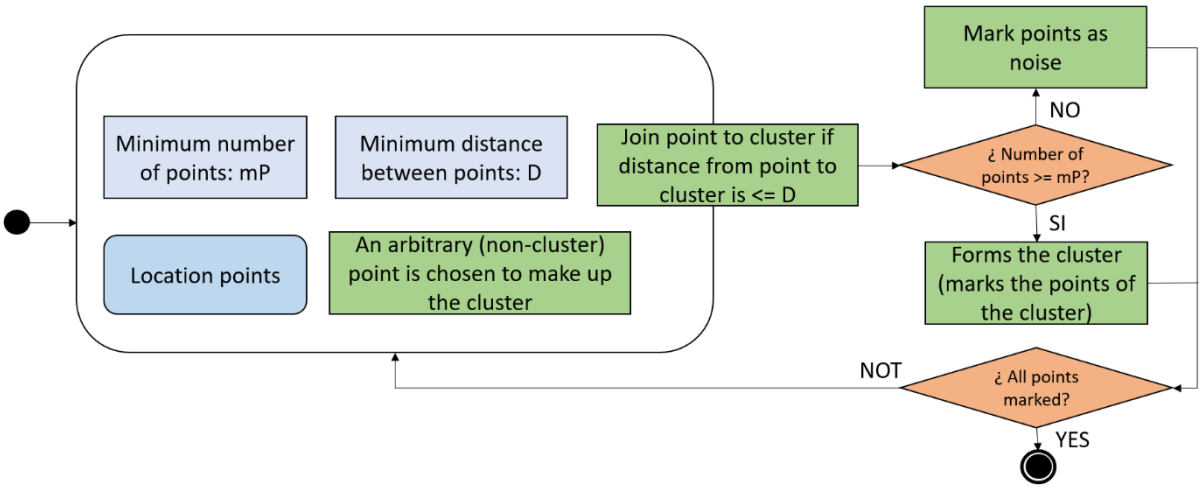

Figure 5 - Clustering process with DBScan

The DBScan algorithm adjusts to the case in which it is not known how many clusters (buses in our example) are needed, but it has the disadvantage that there may be students without an associated group, and the maximum number of cluster members is not controlled.

4.4 Proposed Algorithm: Clustering Routes

To overcome the disadvantages of the previous grouping algorithms, the route group algorithm is proposed, which is intended to select the members of the group that facilitate the traversal of a route. This algorithm starts from the point furthest away from a central point or collection center, and begins to select its members from the point closest to each new member of the cluster, in such a way that it simultaneously defines the order of the route to follow. The tour ends when the maximum size of the cluster is reached, thus it also optimizes the transport resources, and also allows adjusting different capacities of the buses to particular groups.

The algorithm receives as input information the group of points (GP), the central point (PC), the maximum number of members per group (T), the maximum number of clusters (N) and uses the maximum search distance of the next member (D). The algorithm is described below:

==========

Begin Group_Route (***GP, PC, T, N, D***)
J=1, ***G***=1
While there are ungrouped points **AND** ***G*** <= ***N***
i=1, ***d***=0
P = ***GP*** point not marked and farthest from ***PC***
Mark_point (***P, G, i***) # mark point ***P*** in group ***G*** and order ***i***
i = ***i*** + 1

```
        While i <= T AND d <= D
            Q = Point not marked and closest to P
            d = distance (P, Q)
            If d <= D
                Mark_point (Q, G, i)
                i = i + 1
            End If
            P = Q
        End While
        G = G + 1
    End While
Fin Group_Route
```

==========

In the case of the exercise, the algorithm is implemented as a function in PosgreSQL and PostGIS, receiving 250 student locations (yellow dots) as input, see figure 6.

Figure 6 – Position points to be grouped

And we obtain 10 groups of 25 students to generate the routes as shown in figure 7.

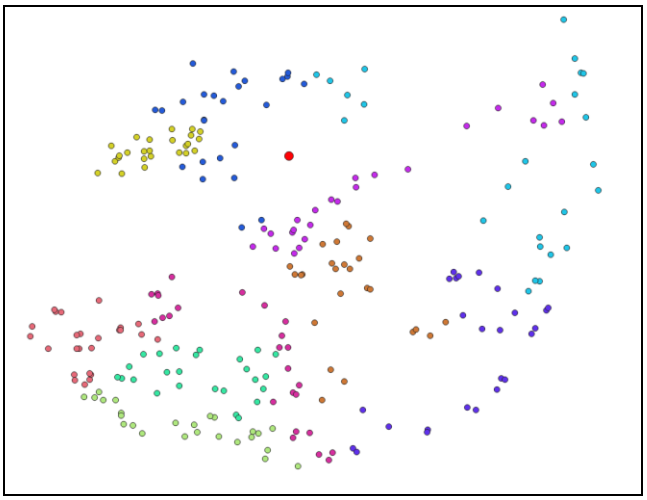

Figure 7 – Grouped position points

In the database, the plate table will have the fields "group" and "num" to indicate the group and the order of travel to generate the route to the school.

The algorithm can easily adapt to the amount of space for each vehicle, if the number of passengers of each vehicle is recorded in the database, this makes the route grouping algorithm easily adaptable to different situations. In case of not knowing the exact number of buses, you can select a large number making the algorithm end when the members run out, or simply if you want to fill the full quota of all buses, select a large number with a maximum distance of member search for the group, so always fill the full quota of each vehicle.

4.2 Route generation - Dijkstra

Once the origin of each of the routes and the subsequent stops for each of the students have been identified, all that remains is to trace the respective routes through the city roads. The algorithm of shortest paths, or Dijkstra's algorithm (Javaid, 2013), is very appropriate for its implementation, since it needs the origin point (initial node), destination point (final node) and a graph, which in our case, is builds from the ways that make up the edges of the graph, and the intersections of the ways make up the nodes of the graph. The edges of the graph need a weight, or value that is taken into account so that the minimum path can be selected, the weight will be the length of the edge.

An algorithm implemented as a procedure in the database is developed, where the information of the routes is initially in table_tracks, the information of the student plates in table_plates, and the information of the school in table_schools. With the information stored in the mentioned tables, the algorithm that builds the routes for each group is described as:

```
=========================================================
Begin Route_Group ()
   Create_table route (gid, geom, cost, group);
   Create_table routes (group, geom);
   g = max_value (group) table_plates;
   For gg=1 to g
      c = count(*) table_plates where grupo=gg
      For nn=1 to c-1
         Insert into table_rout(gid, geom, cost, group) from table_tracks where axis in
            pgr_dijkstra ('gid, start, end, cost from table_tracks',
               (select node from table_plates where group=gg AND n=nn),
               (select node from table_schools where group=gg AND n=nn+1))
      End For
      Insert into table_rout(gid, geom, cost, group) from table_tracks where axis in
         pgr_dijkstra ('gid, start, end, cost from table_tracks',
            (select node from table_plates where group=gg AND n=c),
            (select node from table_school))
         Insert into table_routes
            (select gg, st_union(geom) from table_route where grupo=gg)
   End For
End Route_Group
=========================================================
```

The result of the execution of the procedure will generate all the routes after traversing each group by entering it into the routes table. Each of the routes is built using dijkstra's algorithm from the rgRouting extension (pgRoutingDocumentation.pdf, n.d.) between two student boards. Since the plate is not exactly on the road network, the node of the roads closest to the plate will be chosen. Each section of the route generated is spatially joined to form the route for the group, which is represented with a different color in figure 8.

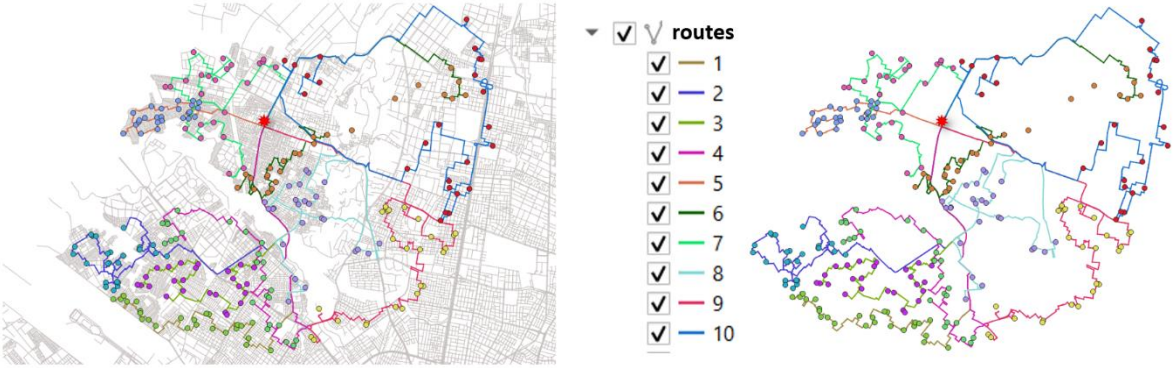

Figure 8 – Routes generated by the routing procedure

5. CONCLUSIONS

The algorithm proposed by route grouping, is very useful as it is adaptable to the number of occupants for each vehicle that makes the route, of course, it can be adapted to any type of capacity, whereas progress is made in the route of the route, the capacity is occupied until it is full. The algorithm adapts to different vehicles with different capacities.

The implementation of the algorithm on a spatial database, allows to have the abstraction of the routes in a graphic way, and the exact calculation of their lengths, in addition to a textual description of the order of the route through the streets of the city. PostgreSQL, being a free database tool, and having extensions for handling spatial information and route calculation, allows the algorithm to be implemented through PL/PgSQL, which facilitates its implementation and use.

The combination of methods and algorithms used has made it possible to adapt the spatial database to the required functionality, allowing solutions to specific problems through different spatial data science techniques. This adaptability makes the database the fundamental basis of an architecture for solving problems that require spatial analysis and computational programming tools to complement the spatial analysis.

REFERENCES

Bação, F., Santos, M., & Behnisch, M. (2020). Spatial Data Science. *ISPRS International Journal of Geo-Information*, *9*, 428. https://doi.org/10.3390/ijgi9070428

Brodie, M. (2019). *What Is Data Science?* (pp. 101-130). https://doi.org/10.1007/978-3-030-11821-1_8

Da, Z.-Y., Yang, W.-J., Ran, P.-P., Qian, X.-G., Shao, S., Pan, S., Shi, B.-J., Li, Y., He, R.-L., & Xiao, Y.-H. (2017). Design of Logistics Route Planning for Printing Enterprises Based on Baidu Map. *ITM Web of Conferences*, *11*, 10001. https://doi.org/10.1051/itmconf/20171110001

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise*. 226-231.

Honduvilla, I., & Manso Callejo, M. Á. (2010). *Servicio web de Geocodificación para Cartociudad*.

Infraestructura de Datos Espaciales de Bogotá. (s. f.). Recuperado 21 de octubre de 2018, de https://ideca.gov.co

Javaid, A. (2013). Understanding Dijkstra Algorithm. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2340905

Kubat, M. (2017). *An Introduction to Machine Learning*.

McDonald, Y., Schwind, M., Goldberg, D., Lampley, A., & Wheeler, C. (2017). An analysis of the process and results of manual geocode correction. *Geospatial Health*, *12*. https://doi.org/10.4081/gh.2017.526

Morissette, L., & Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology*, *9*, 15-24. https://doi.org/10.20982/tqmp.09.1.p015

Owusu, C., Lan, Y., Zheng, M., Tang, W., & Delmelle, E. (2017). *Geocoding Fundamentals and Associated Challenges* (pp. 41-62). https://doi.org/10.1201/9781315228396-3

PgRoutingDocumentation.pdf. (s. f.). Recuperado 15 de septiembre de 2022, de https://docs.pgrouting.org/2.0/es/pgRoutingDocumentation.pdf

Shalev-Shwartz, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*.

Wilson, J., & Knoblock, C. (2007). From text to geographic coordinates: The current state of geocoding. *Urisa Journal*, *19*, 33-46.

CONTACTS

Alvaro Enrique Ortiz Dávila
UniversidadDistrital Francisco José de Caldas
Carrera 7 # 40B-53
Bogotá D. C.
COLOMBIA
Tel. +57 3106792857
Email: aeortizd@udistrital.edu.co

Luz Angela Rocha Salamanca
UniversidadDistrital Francisco José de Caldas
Carrera 7 # 40B-53
Bogotá D. C.
COLOMBIA
Tel. +57 3106792857
Email: lrocha@udistrital.edu.co