

Shortest path algorithms on GIS dataset using geotools

¹Richard Somalia, ²Dr. M.B. Potdar, ³Manoj Pandya, ⁴Rupesh Punjani, ⁵Neeraj Bhargava

Abstract— GIS or Geographic Information System has the ability to query and analyze geographic information in a variety of different contexts. .GIS is a technology which is ideally suited for analysis of the market values of properties, since such values are based upon spatial comparisons as well as individual property attributes. It is well known that computing shortest paths over a network is an important task in many network and transportation related analysis. Choosing an adequate algorithm from the numerous algorithms reported in the literature is a critical step in many applications involving real road networks. We are applying a shortest path algorithm on a graph generated from a shape-file's feature of road network with the distance as the weight-age of a graph node and displaying shortest path with its node information in this tool.

Index Terms— shortest path algorithms, Geographical information system (GIS), Dijkstra algorithm, shape-file, geotools, graph, maven eclipse

1 INTRODUCTION

With the development of geographic information systems (GIS) technology, its analysis power is applied in many fields of real world problems. A mathematical model of any network or 3D representation of objects is possible with the consideration of different forces. We can apply GIS to calculate the shortest path algorithms on .SHP file. The Esri shape-file, or simply a shape file, is a popular geospatial vector data format for geographic information system software. It is developed and regulated by Esri as an open specification for data interoperability among Esri and other GIS software products. Shape-files spatially describe vector features: points, line and polygons, representing, for example, water body, well, rivers, and lakes etc. Each item usually has attributes that describe it, such as name or temperature [1]. One can apply shortest path algorithms like Dijkstra on Shape-file file with the help of GEOTOOLS (osgeo) library available in Java. GEOTOOLS can be used with the help of MAVEN. Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information. Maven uses an XML file to describe the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging [2]. With the help of this kind of model, we can select direct source point to destination point with its node and attributes information in a real time scenario.

- Richard Somalia is affiliated with JNTU, Hyderabad, e-mail: richisonaliya@gmail.com
- Dr. M.B. Potdar is currently working as project director, BISAG
- Manoj Pandya is currently working as a project Manager at Bhaskaracharya Institute for Space Applications and Geo-Informatics (BISAG), Gandhinagar, Gujarat, India PIN 382007 E-mail: mjpandya@gmail.com
- Rupesh Punjani is working as a consulting service lead, IBM
- Dr. Neeraj Bhargava is affiliated with MDS University, Ajmer, Rajasthan

2 SHORTEST PATH USING DIJKSTRA

Dijkstra's algorithm was developed by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959. It is based on graph search, the edge and vertex, gives the shortest path between two vertex. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined [3]. The algorithm is represented in brief as below [4].

$G = (V, E)$ where V is a set of vertices and E is a set of edges.

Dijkstra's algorithm keeps two sets of vertices:

S the set of vertices whose shortest paths from the source have already been determined

$V-S$ the remaining vertices.

The other data structures needed are:

D array of best estimates of shortest path to each vertex

P_i an array of predecessors for each vertex

2.1 The basic mode of operation is:

1. Initialise d and p_i ,
2. Set S to empty,
3. While there are still vertices in $V-S$,
 - i. Sort the vertices in $V-S$ according to the current best estimate of their distance from the source,
 - ii. Add u , the closest vertex in $V-S$, to S ,
 - iii. relax all the vertices still in $V-S$ connected

to u
2.2 Pseudo code

```

dist[s] ← 0           (distance to source vertex is zero)
for all v ∈ V - {s}
    do dist [v] ← ∞   (set all other distances to infinity)
S ← ∅                 (S, the set of visited vertices is
initially empty)
Q ← V                 (Q, the queue initially contains all
vertices)
while Q ≠ ∅           (while the queue is not empty)
do u ← mindistance( Q, dist ) (select the element of Q with the
min. distance)
    S ← S ∪ {u}       (add u to list of visited vertices)
for all v ∈ neighbors[u]
    do if dist [v] > dist [u] + w(u, v) (if new shortest path
found)
        then d[v] ← d[u] + w(u, v) (set new value of shortest
path)
            (if desired, add tracebackcode)
return dist
    
```

3 USING OPEN SOURCE GeoTools

GeoTools is a free software GIS toolkit for developing standards compliant solutions. It provides an implementation of Open Geospatial Consortium (OGC) specifications as they are developed. GeoTools is a contributor to the GeoAPI project - a vendor-neutral set of Java interfaces derived from OGC specifications - and implements a subset of those libraries.

A Shapefile is a common file format which contains numerous features of the same type. Each shapefile has a single feature type [5].

The classic three files:

- filename.shp: shapes
- filename.shx: shapes to attributes index
- filename.dbf: attributes

Basic metadata: * filename.prj: projection

Open source extensions:

- filename.qix: quadtree spatial index
- filename.fix: feature id index
- filename.sld: style-layer-descriptor style xml object

ESRI extensions:

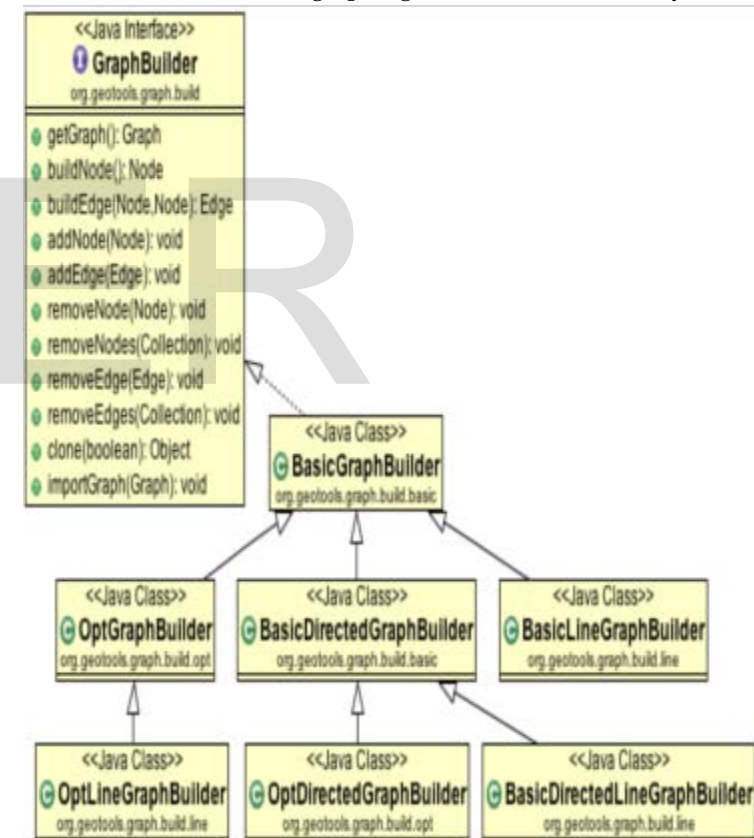
- filename.sbn: attribute index
- filename.sbx: spatial index
- filename.lyr: arcmap-only style object
- filename.avl: arcview style object
- filename.shp.xml: fgdc metadata

The gt-graph package defines the concept of a graph (or network) made up of .shp file Features. The graph provides

flexible way to make computation in directed and undirected networks. For a given road network we have to create a graph of that road network with the distance as their weightage. The shape-file is a vector data file containing the feature and its attributes where graph is having edges and nodes. So we have to take a Hash Map to map the feature with its respective edge in the graph. To select a point on a shape-file we are creating a filter of 5x5 pixel square .selected feature's node are passed to the algorithms to calculate the shortest path. The path is calculated on the basic of the distance. It also represents the node visited during the graph traverse.

3.1 METHODOLOGY

The methods of the class used to convert shape-file into graph are shown as below. A Graph-Builder is used to build and retrieve information from shape-file [6]. The graph module makes the use of Feature, FeatureType, FeatureID. At low level, graph generation is done by



GraphBuilder.

Fig-1 the class diagram to generate graph

getGraph() method is used to retrieve the generated graph. BasicLineGraphGenerator() is used to build a line network. With the help of different types of GraphGenerator, one can create directed or undirected graph as per the need.

4 CODE SNIPPET FOR DIJKSTRA

We have a number of ways to calculate the shortest path between two nodes. The class DijkstraShortestPathFinder can be used to calculate a path using Dijkstra's Shortest Path algorithm [7].

```
DijkstraIterator.EdgeWeighter weighter = new DijkstraItera-  
tor.EdgeWeighter() {  
    public double getWeight(Edge e) {  
        SimpleFeature feature = (SimpleFeature) e.getObject();  
        Geometry geometry = (Geometry) fea-  
ture.getDefaultGeometry();  
        return geometry.getLength();  
    }  
}
```

```
// Create GraphWalker - in this case DijkstraShortestPath-  
Finder  
DijkstraShortestPathFinder pf = new DijkstraShortestPath-  
Finder( graph, start, weighter );  
pf.calculate();
```

```
//find some destinations to calculate paths to  
List/*<Node>*/ destinations = destination_node;
```

```
//calculate the paths  
for ( Iterator d = destinations.iterator(); d.hasNext(); ) {  
    Node destination = (Node) d.next();  
    Path path = pf.getPath( destination );
```

```
//do something with the path  
}
```

5 EXPERIMENTS AND RESULT

Shortest path algorithm has been applied on GIS dataset with the help of geo-tools open source libraries. USA road network shape-file is referred. Its attributes and projection information is populated in a tabular format. A graph which represents shape-file features with distance as its weightage is generated. A hash map is constructed to store feature id and the graph edges. A 5x5 pixel filter is used to select a source point and a destination point to calculate a

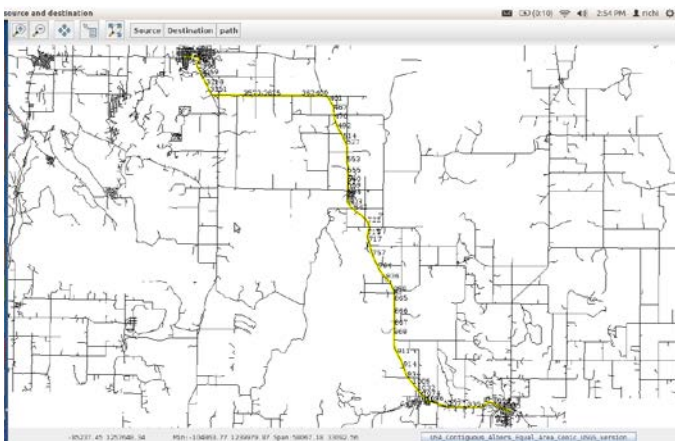


Fig-2 way of traverse between two selected nodes

path in a real time. The model highlights the shortest path in a yellow colour between selected two nodes in a sequential way of traverse.

6 EXTENSION OF WORK

The model requires longer period of time to calculate the shortest distance for a larger GIS road network dataset. To eliminate this issue, GIS network should be preprocessed. Preprocessed output of the graph generation can be used in this model. BOINC can be used to generate the graph on the server with the help of distributed computing [5]. The Berkeley Open Infrastructure for Network Computing (BOINC) is an open source middleware system for volunteer and grid computing. BOINC is designed to be a free structure for anyone wishing to start a volunteer computing project. Most BOINC projects are nonprofit and rely heavily, if not completely, on volunteers. Graph generation process can be done on the server with the help of BOINC.

7 CONCLUSION

The demand for GIS network analysis is increasing day by day. Open source geotools libraries and BOINC for preprocessing of GIS dataset can be used in distributed computation environment to improve efficiency.

8 ACKNOWLEDGEMENTS

The authors would like to thank to the Director, BISAG, T.P. Singh for his inspiration and motivation. Authors would also like to thank to Mr. Abdul Zummerwala for technical support.

8 REFERENCES

- 1 "Esri Shape-file for GIS software as vector data format", Internet : <http://en.wikipedia.org/wiki/Shapefile> .[11 aug 2013]
- 2 "Apache Maven Introduction", Internet: <http://maven.apache.org/> [10 aug 2013]
- 3 "Dijkstra algorithm", Internet : http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm [10 aug 2013]
- 4 "Dijkstra algorithm flow and pseudo-code for shortest path calculation", Internet : <http://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html>. [11 aug 2013]
- 5 "Shape-file", Internet: <http://docs.geotools.org/latest/userguide/library/data/shape.html> [10 aug 2013]
- 6 "geotools", Internet : <http://docs.geotools.org/latest/userguide/extension/graph/index.html>, [5 aug 2013]
- 7 "geotools graph", Internet : <http://docs.geotools.org/latest/userguide/extension/graph/index.html>. [15 aug 2013]
- 8 Kang-Tsung-Chang "Introduction to GIS", Tata McGraw Hill, 4th ed., 2008.

- 9 Heywood Ian ,Sarah Cornelius ,Steve Carver“ An Introduction To Geographical Information Systems “, Pearson ,3rd ed,2010.

IJSER