Search Space Reduction using K-means clustering and Adjacency matrices for GIS-usage Information Retrieval

Wai Mar Hlaing, Myint Myint Sein
University of Computer Studies, Yangon
waimarhlaing16@ gmail.com, myintucsy@gmail.com

***Abstract* - Nowadays, people widespread use of smartphones and ubiquitous devices for map based services. As the transport network is complicated and massive, people may be confused to reach the desired location after finding a location. Many searching techniques are used for finding the shortest path, might still not be fast enough in certain real-time applications because of complexing transport network. Search time can be reduced if we pruned unnecessary clusters in a complex large graph. Memory utilization is safe for the processing time if we reduce search space in complex network. For removing unnecessary clusters, adjacency matrices, distance based methods and K-means clustering can be used. ArcGIS software and popular shortest path algorithms are applied to find the shortest path from one location to another on the Android mobile platform. In addition, the performance of finding the shortest path using popular A* and Dijkstra algorithms with bidirectional search can be compared before and after removing unnecessary clusters.**

***Keywords* – K-means clustering algorithm, Adjacency matrix, Distance based methods, A* with bidirectional, Dijkstra with bidirectional, ArcGIS**

I. INTRODUCTION

Geographic information system has been used in several areas such as transportation, emergency services, and health care planning. Finding the shortest path problem is always encountered in our daily life. Many shortest path algorithms can be applied in finding the shortest path on the geographic data. Many geographic information systems have been applied using the popular shortest path algorithms intending the high-speed performance. Among of the shortest path algorithms, A* is the most suitable for the systems that know the source and destination. Dijkstra algorithm is the most appropriate for the blind search and the small number of objects. Among of the shortest path algorithms, A* and Dijkstra algorithms are used together with bidirectional search algorithm for receiving the high-speed performance in finding the complex network. However, fast enough searching problem can still occur because the entire search space for the complex network based on map-based search such as GPS navigation and robotics is very large .So, search space reduction is essentially needed to explore exponentially large combinatorial problems. Data Mining is the used for the many areas. Association, classification and clustering are the popular techniques in data mining. In this paper, K-means clustering, distance based method, and adjacency matrices are used for search space reduction in the complex network at the pre-processing step. Adjacency matrices, adjacency list and many other techniques can be used to satisfy the storage efficiency for the graph data structure.

II. LITERATURE REVIEW

Amrapali Dabhade, Dr. K. V. Kale and Yogesh Gedam in 2015, works the network analysis for finding shortest path in hospital information systems [1]. In this situation, sometimes it is not easy to find the specialized hospital and its shortest path to reach the desired location hence takes more time to reach it. This paper tries to solve the problem in finding shortest paths facility for getting the nearest location of the hospitals from user's current location[1]. Yizhen Huang, Qingming Yi, Min Shi in 2013, presented an improved Dijkstra shortest path algorithm [2].This algorithm introduces constraint conditions for searching each position in the state space to guide the search forward to expected direction. In this , the weighted value is flexibly changed to adapt to different network complexity. By introducing constraint function, it can omit lots of useless search path and accelerate the process of problem-solving. In this case, the optimal solution can be found using the proposed algorithm and can improve the search efficiency. Parateek Jogjon compares the popular shortest path algorithms and Dijkstra Algorithm [3].

III. SYSTEM OVERVIEW

In this system, this contains three main parts. These are (i)Data Collection,(ii)Pre-processing and (iii)Finding the shortest path and giving the desired information on the map as shown in Fig .1..Firstly, in the data collection stage, we must collect the required data and this data must be transformed into a digitized form for the computerized processing. Secondly, our proposed system will be pre-processed for removing the unnecessary clusters and reducing the search space at the stage of finding the shortest path. Finally, after pre-processing stage is implemented, our system will find the shortest path on the complex networked graph using the popular shortest path algorithms such as A* and Dijkstra algorithms. To improve the performance for the complex graph, A* algorithm and Dijkstra algorithm will be used together with bidirectional algorithm.

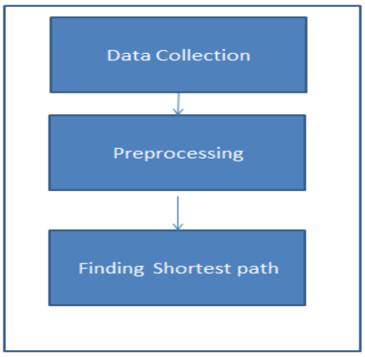


Fig. 1 Overview of the proposed system

IV. Data Collection

Data collection is an important aspect of any type of research study. Inaccurate data collection can impact the results of a study and ultimately lead to invalid results. Our system is intended for retrieving the GIS usage information in fast speed. In this system, geo-database is constructed manually. The required geodata for the complex road network of the Yangon Region can be acquired as KML file from google map and nations online. This data also gets from Toposheet. Data collection stage is implemented as shown in figure(2). Digitizing converts paper map features into digital format. In this step the data for the road network is extracted from Toposheet and KML file[4]. In the objects mapping, the coordinates of the objects is taken by doing the field survey. And based upon that coordinates of the different objects such as restaurants, hotels, and interesting places are placed on the map as a point feature. Geo-database has been constructed by combining the actual positions of different objects (i.e. longitude and latitude values taken GPS) using the collected data objects.

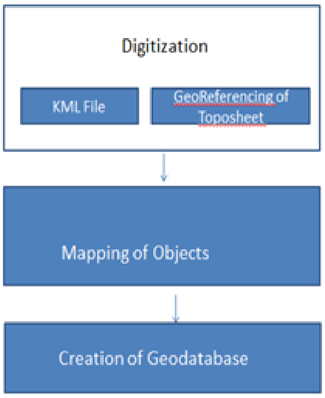


Fig. 2 The process flow of the data collection stage

V. Preprocessing

After the data collection is finished, this geo-database is applied for the search space reduction at the pre-processing stage. Huge search space is usually difficult and complex to retrieve the GIS usage Information. So, we proposed the search space reduction algorithm for the complex networked graph. Our proposed algorithm contains three main steps for the search space reduction as shown in Fig. 3. Firstly, the proposed algorithm implements the adjacency matrices for the data structure. After that, these data are clustered using K-means clustering algorithm. And then, the clusters that do not contain source and destination are removed.

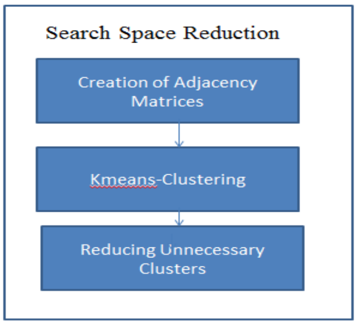
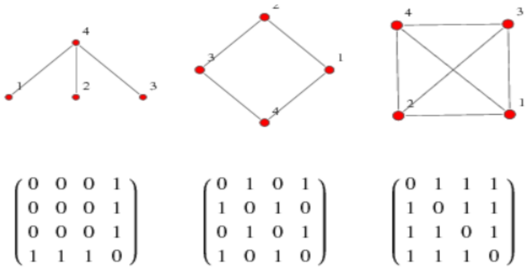


Fig. 3 The process flow of the pre-processing stage

A. Adjacency Matrix

The adjacency matrix as shown in Fig .4, sometimes also called the connection matrix, of a simple labelled graph is a matrix with rows and columns labelled by graph vertices , with a 1 or 0 in position (v_i, v_j)according to whether vi and vj are adjacent or not. For a simple graph with no self-loops, the adjacency matrix must have 0s on the diagonal. For an undirected graph, the adjacency matrix is symmetric [5].



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Fig. 4 Graph and Adjacency Matrix

B. Distance Based Methods

The distance between the start node and destination node are computed in advance using distance base methods. These data are stored by using adjacency matrices for different types of objects. There are many distance based methods [6]. These are

(1)Chebyshev distance-measures distance assuming only the most significant dimension is relevant.

(2)Euclidean distance matrix

(3)Hamming distance-identifies the difference bit by bit two strings

(4)Mahalanobis distance-normalizes based on a covariance matrix to make the distance metric scale-invariant.

(5)Manhattan distance-measures distance following only axis-aligned directions.

(6)Metrics- Distance matrix, Hierarchical clustering

(7)Minkowlsi distance is a generalization that unifies Euclidean distance, Manhattan distance, and Chebyshev distance.

(8)Pythagorean addition

(9)Haversine distance-giving great-circle distances between two points on a sphere from their longitudes and latitudes.

(10)Vincenty’s formula well known as “Vincent distance.

C. K-Means Clustering

Adjacency matrices are applied using K-means algorithm for removing unnecessary clusters. K-means is one of the simplest unsupervised learning algorithms that solve the well -known clustering problem [7]. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centres, one for each cluster. These centres should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centre. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycentre of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centre. A loop has been generated. As a result of this loop we may notice that the k centres change their location step by step until no more changes are done or in other words centres do not move any more. Finally, this algorithm aims at minimizing an objective function knows as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} (||x_i - v_j||)^2$$

where, *'*$||x_i - v_j||$*'* is the Euclidean distance between x_i and v_j.
'c_i*'* is the number of data points in i^{th} cluster.
'c' is the number of cluster centres.

1) Algorithmic steps for K-means clustering

Let X={x1,x2,x3,…..,xn} be the set of data points and V={v1,v2,……,vc} be the set of centres.

1. Randomly select 'c' cluster centres.
2. Calculate the distance between each data point and cluster centres.
3. Assign the data point to the cluster centre whose distance from the cluster centre is minimum of all the cluster centres.
4. Recalculate the new cluster centre
5. Recalculate the distance between each data point and new obtained cluster centres.
6. If no data point was reassigned then stop, otherwise repeat from step 3).

After finding the clusters for the complex graph, the proposed system will remove unnecessary clusters. Unnecessary clusters mean the clusters that do not contain source and destination nodes.

VI. FINDING THE SHORTEST PATH

After pre-processing is made, we can find the shortest path on the road network more easily together with high-speed performance. In this paper, we use the A* and Dijkstra algorithm together with binary search for improving the performance.

A. A* Algorithm

A* algorithm is a graph search algorithm that finds a path from a given initial node to a given goal node [8]. It employs a "heuristic estimate" h(x) that gives an estimate of the best route that goes through that node. It visits the nodes in order of this heuristic estimate. It follows the approach of best first search. The secret to its success is that it combines the pieces of information that Dijkstra's algorithm uses (favouring vertices that are close to the starting point) and information that Best-First-Search uses (favouring vertices that are close to the goal). In the standard terminology used when talking about A*, g(n)represents the exact cost of the path from the starting point to any vertex n, and h(n) represents the heuristic estimated cost from vertex n to the goal.

B. Dijkstra Algorithm

Dijkstra's algorithm is a graph search algorithm that solves the single source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. 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. The working of Dijkstra algorithm can be viewed as the following.

```
distance[source] ←0
 for all vertex v εV–{source}
do distance[v] ←∞ SET←ϕ (S, the set of visited vertices is
initially empty)
 QUEUE←V (Q, the queue initially contains all vertices)
 while QUEUE ≠ ϕ
do
            u ← min distance(QUEUE,dist)
             SET←SETʊ{u} for all vertex,
            v ε neighbors[u]
 do
             if distance[v] > distance[u] + weight(u, v)
            then
            distance[v] ←distance[u] + weight(u, v)
return distance[ ].
```

C. A* and Dijkstra algorithms with Bidirectional Search

Bidirectional search is a graph search algorithm that finds a shortest path from an initial vertex to a goal vertex in a directed graph. It runs two simultaneous searches: one forward from the initial state, and one backward from the goal, stopping when the two meet in the middle. The reason for this approach is that in many cases it is faster. In this paper, A* algorithm and Dijkstra algorithm together with Bidirectional search will be used for the shortest path finding to get the high speed access for the GIS usage Information retrieval. So, two directional searches will be used in these algorithms concurrently.

VII. EXPERIMENTAL RESULTS

System will generate the shortest path between two locations by calculating the distance based on road length. This will help the user to reduce the travelling time to reach a desired location. In this system we compare the processing time before and after our proposed pre-processing step are used. In Fig. 5, the processing time is compared before and after pre-processing step is used by applying the Dijkstra with Bidirectional. In Fig. 6, the processing time is also compared before and after pre-processing step is used by applying the A* with Bidirectional. In this system, Android Sdk is used for creating the GPS based android mobile application. 30 townships in Yangon Region are used as the spatial database for implementing the proposed search reduction technique.

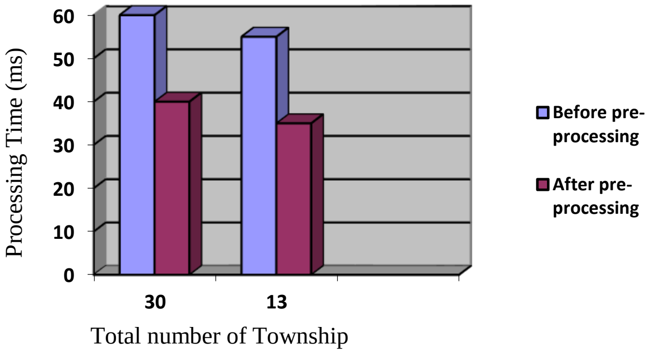


Fig 5.Performance comparison between the processing time before and after pre-processing step is used for Dijkstra's Algorithm with Bi-directional

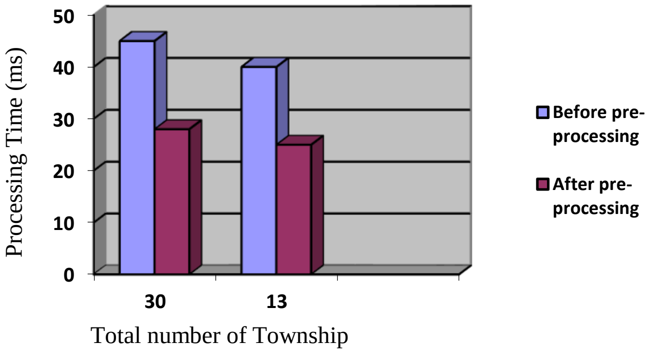


Fig 6.Performance comparison between the processing time before and after pre-processing step is used for A* Algorithm with Bi-directional

VIII. Conclusion

This research retrieves the Gis usage information such as the shortest path from the user location or selected source to the required destination on the complex network. It uses the Android Mobile ArcGIS, Android SDK, A* algorithm, the Dijkstra's algorithm with bidirectional search and K-means clustering algorithm are used for implementation. ArcGIS is used to digitize the map into road network. Data structures and distance base methods are used for applying the K-means clustering and removing the unnecessary clusters for the search space reduction. In addition, we can compare the searching time in finding the shortest path between the applying to the proposed search reduction method and without using it. The system will help to retrieve GIS usage information in high-speed by using the search reduction technique.

REFERENCES

[1] Amrapali Dabhade, Dr. K. V. Kale, Yogesh Gedam,"Network Analysis for finding Shortest Path in Hospital Information System", IEEE,vol 5,July 2015

[2] Yizhen Huang, Qingming Yi, Min Shi, "An Improved Dijkstra Shortest Path Algorithm",ICCSEE 2013

[3] Abhishek Goyal, Prateek Mogha, Rishabh Luthra, Ms.Neeti Sangwan, "Path Finding:A* or Dijkstra;s?" IJITE,vol4,January,2014

[4] ArcGIS information accessed on Oct 2013.[Online]Available :http://resources.arcgis.com/en/help/gettingstarted/articles/026n00000014000000.html

[5] WorldFarm MathWorld website. [Online].Available: http://mathworld.wolfram.com/AdjacencyMatrix.html

[6] Wikipedia website .[Online].Available: https://en.wikipedia.org/wiki/Euclidean_distance

[7] Data Clustering Algorithms website.[Online].Available: https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm

[8] Wikipedia website .[Online].Available: https://en.wikipedia.org/wiki/A*_search_algorithm

[9] Roozbeh Shad, Hamid Ebadi, Mohsen Ghods. "Evaluation of Route Finding Methods in GIS Application", International Journal of Computer Applications (0975 – 8887) Volume 53– No.10, September 2012.