CHAPTER 5

HIERARCHICAL COMMUNITY MINING – FUZZYANT DYNAMIC ROUTING ON LARGE ROAD NETWORKS

5.1 INTRODUCTION

After the earlier phase of three chapters in which the founding principles and data for the thesis are presented, the second phase of the thesis is initiated. This chapter is the first of a series of three chapters in which fuzzy algorithms are introduced. The pragmatic user-based route selection on large road networks.

Route selection is an important decision in road networks. The problem confronts not only the city dwellers but also the boulevards. Hence a study is initiated towards a hierarchical community mining system that facilitates the development of a model road network statically. This study gets strengthened with the introduction of a dynamic technique for planning a fast route planning in large road networks. This is relatively a new approach that generalizes and combines fuzzy logic for local pheromone and an updated ant colony system in the identification of optimum multi parameter direction between two desired points, origin and destination. The significance of the hierarchical community based routing algorithm appreciably reduces the search space. The next step attempted is the extension of a novel Hierarchical Community Mining approach adopting Fuzzy Logic Ant Colony
System (FLACS). This scheme is quite distinctive and professional because it could support the dynamic and efficient route computation on large road networks.

Thus we find that computing the fastest routes in road networks has become an enthusiasm among researchers in real world applications. The algorithm that could aid in finding the shortest path is provided by Dijkstra (1959). His approach becomes very slow when applied to large road networks. There are other techniques that are comparatively faster and more practical in route planning. Sanders &Schultes (2007) the most widespread and triumphant methods are static in assumptions i.e. they presume that the networks and the edge weights do not change. But the reality is contrary in truth. The computational effort is still quite high as the network size becomes larger, the real world routing plan also changes accordingly. Hence, it has been planned to take up both hierarchical approach and the dynamic setting.

5.2 HIERARCHICAL APPROACH

As has been observed earlier, the computational attempt becomes more and more complex as the network size becomes sizably large and this makes the scheme unsuitable for real time computing. To avoid this, there is a possibility of precomputing and storing all pairs of shortest paths in a distance table. But this endeavour is a daunting and impractical one since a huge area of storage would be required which, in turn, is bound to exceed the memory limit, when large road maps are considered. A better design would be to precompute and store some helpful information Chen et al (2007). Hierarchical approach is undertaken in order to take hold of some important vertices and arcs in road networks with the employment of community mining algorithm.
5.3 DYNAMIC ENVIRONMENT

The very intent behind the finding of the shortest path is the cost factor, the route which is available at the least cost, obtained through the calculation of costs for different routes. Most of the past research is devoted to optimum route selection relying on several significant and relevant parameters. But these are related to static scenario only. But more and more concerns emerge in the selection of routes such as fuel cost and delay due to traffic jams and congestions causing enormous loss of time and energy. All these have been elaborately dealt with in the earlier chapter on risk factor classification. The road users do not remain satisfied with the resolution of the above factors alone. They not only prefer routes that are short but also fulfill some of their needs also. These needs may include safety, low density in traffic, scenic environment and fewer road crossings / junctions to facilitate smooth and continuous movement with traffic signals.

5.4 METHODOLOGY

The technique adopted is a combination of fuzzy logic and ant colony system described by Salehinejad&Talebi(2010). This format is adopted to find the optimum multiparameter route between a source and destination, the optimum route that attempts to satisfy all desired parameters of a user, the individual edge weight updates based on traffic, quality, distance, road condition and toll gate charge.

5.5 HIERARCHICAL COMMUNITY – FUZZY ANT BASED ROUTING

Figure 5.1 is an illustration of the stages in the development of hierarchical community fuzzy-ant based dynamic routing algorithm for large road networks architecture.
Figure 5.1 Flowchart of hierarchical-fuzzy ant based routing
### 5.5.1 Hierarchical Community Algorithm

The hierarchical community algorithm consists of two phases.

- Distance based community phase
- High level community phase

This finds exemplification in Figure 5.2.

<table>
<thead>
<tr>
<th>Input:</th>
<th>Graph G with edge Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>Number of Sub graph</td>
</tr>
</tbody>
</table>

**Partitioning phase:**

1. Get a region value as user input
2. Initiate with starting node
3. For each vertex
4. Calculate the total edge weight for connected edge
5. If total weight $\leq$ user's region value
6. Add into one group
7. Else other group
   
   End For

**High level community phase:**

1. Find the border node based on colored vertex
2. Add the so-called community edge between every pair of border nodes.
3. With-in community edges denoted by dashed line
4. Between community edges denoted by solid lines.

*Figure 5.2 Distance based community formation*
The partitioning phase is used to split a given network \( G = (V, E, W) \) into sub graph based on Neighborhood Random Walk Distance definitions and high level community phase is used for constructing \( G^P \) from the sub graph \( G_u \). Community phase is based on user’s region value. Total weight is to be calculated and checked to find whether it is less than the region value. If it is discovered to be less, \( G_u \) group is added; otherwise \( G_v \) group is added.

In High level community phase, the border node based on color vertex is selected. It is commonplace that the same colored vertex forms homogenous group. For any node \( i \in V_u \), if there exists a node, \( j \in ADJ(i) \), \( i \) colored vertex \( \neq j \) colored vertex, then ‘\( i \)’ is named border node of \( G_u \) and ‘\( j \)’ is named border node of \( G_v \). The inter community edges are denoted by a solid line. The so-called community edges are added between every pair of border nodes of each sub graph and this one is denoted by a dashed line. So these edges construct the high level community graph.

### 5.5.2 Hierarchical Graph Model

An extensive variety of processes have been raised for detecting hierarchical community in networks (Fortunato 2010). Of these, distance based community detection algorithm is chosen for the present study. The basic idea of community detection algorithm involves a number of definitions Essam& Fisher(1970). A network can be modeled as a graph \( G = (V, E, W) \) where \( V \) is the set of nodes and \( E \) is the set of links and \( w \) is the set of edge weights. The Graph \( G \) is partitioned into \( p \) communities at level \( l \), with each community corresponding to a sub graph \( G_{l'} = V_{l'}, E_{l'}, W_{l'} \in G \).

A Partition of \( P = \{G_1, G_2, \ldots, G_p\} \) is of \( G \). Two sub graphs \( G_{l'}^1, G_{l'}^2 \) is said to be adjacent if \( G_{l'}^1, G_{l'}^2 \) is an edge of \( G \). If the given partition
\( P = \{G_1^1, G_2^1, \ldots, G_b^1\} \) is of \( G \) and the edges that link adjacent sub graphs \( G_{i'j'} \) and \( G_{i'j'} \) are called the intercommunity edge set, the set is denoted by

\[
\text{INTERCOM}(G_{i_1}^1, G_{i_2}^1) = \left\{ (i, j) \middle| (i, j) \in E \land (i \xrightarrow{c(i,j)} j \text{ in } G) \land (i \in \text{BORDER}(G_{i_1}^1)) \land (j \in \text{BORDER}(G_{i_2}^1)) \right\}
\] (5.1)

The intercommunity edges can be forming the bottlenecks between sub graphs. The cost function \( f_c(i, j) \) gives the cost of the shortest path from node \( i \) to \( j \).

### 5.6 COMMUNITY GRAPH

A sub graph is by and large characterized as a community. The community edge set is defined in terms of the following formula.

\[
\text{COMU}(G_{i_1}^1) = \left\{ (i, j) \middle| (i, j) \in \text{BORDER}(G_{i_1}^1) \land (i \xrightarrow{c(i,j)} j \text{ in } G) \land (i \neq j) \right\}
\] (5.2)

The next step is the addition of the assumed community edges between every pair of border nodes. Finally, all the adjacent sub-groups are linked through inter community edges.

#### 5.6.1 Community Detection Phase

A community within a network is a group of vertices densely glued to each other but less connected to the vertices outside. Vertices tend to organize themselves in groups (called communities or clusters) in such a manner that the intersections that are located close in small regions are more likely to form a community. The network is then decomposed, with adjacent
sub networks being loosely connected by the intergroup edges. In this approach, each sub network forms an isolated part and different parts are connected through boundary or border nodes. All the shortest paths between different communities should go along one of these few edges. The approach implemented in this chapter is a distance estimation mechanism Blondel et al(2008). The optimality of the algorithm is guaranteed in one of the many following practices.

5.6.2 Structural Closeness Measure

In a large graph $G$, some vertices are close to each other while some other vertices are far apart depending upon connectivity. If there are multiple paths connecting two vertices $vi$ and $vj$, then they are presumed close. On the other hand, if there are very few or no paths between $vi$ and $vj$, then they are considered to be far apart. Neighborhood random walk distances have been taken into consideration to measure vertex closeness.

5.6.3 Neighbourhood Random Walk Distance

Let $P$ be the $N \times N$ transition probability matrix of a graph $G$. Given $l$ as the length that a random walk can go, $c \in (0, 1)$ as the restart probability, the neighborhood random walk distance $d(vi, v j)$ from $vi$ to $vj$ is defined as

$$d(vi, v j) = \sum_{\tau : \text{length}(\tau) \leq l} p(\tau) c \left(1 - c\right)^{\text{length}(\tau)}$$  \hspace{1cm} (5.3)$$

where $\tau$ is a path from $vi$ to $vj$ whose length is $\text{length}(\tau)$ with transition probability $p(\tau)$.

The matrix form of the neighbourhood random walk distance is
\[ R^l = \sum_{y=1}^{l} c(1 - c)y^P \]  

(5.4)

\[ d_S(v_i, v_j) = R^l(i, j) \]  

(5.5)

5.7 HIGH LEVEL COMMUNITY PHASE

The border node is selected derived from color vertex in high level community phase. It is the universal knowledge that the similar colored vertices form a single group. The inter community edges are indicated with solid lines. The supposed community edges between every node of border nodes of each sub group, specified by the dashed lines, are added. These edges constitute the high level community graph.

5.7.1 Fuzzy Ant Based Algorithm

The inquiry on fuzzy-ant based routing is developed using a fuzzy logic technique to solve the network routing problem Mirabedini et al(2008). The scrutiny of multiple constrictions becomes simple and spontaneous in the process.

5.7.2 Ant Based Routing

Ant based routing is a meta-heuristic algorithm. It connotes that it’s a general framework that facilitates the creation of an individual algorithm to solve an exclusive graph path problem. Although ant based routing was proposed in a doctoral thesis by Dorigo et al(1991), the first detailed description of the algorithm is generally attributed to a 1996 follow-up paper published in Dorigo et al (1996). Ant based routing algorithms are the
applications of the knowledge stemming from recent perception of basic
principles underlying the operation of swarms, often containing thousands of
elements, routinely performing extraordinarily complex tasks of global
optimization and resource allocation built on local information alone. These
properties make ant colony very attractive for network routing by Sim &

5.8 FUZZY LOGIC FOR ANT BASED ALGORITHM

Fuzzy ant based algorithm is raised from the interesting observation
of the concurrent and concomitant large road network flourishing in ant
colonies combined with the capabilities of the fuzzy logic technique. This
algorithm first determines the crisp path ratings for all eligible paths between
the source and destination nodes from the viewpoint of fuzzy inference. The
path with the highest rating is then chosen to route the shortest path. The
fuzzy inputs are chosen as the ‘traffic,’ ‘road condition,’ ‘quality,’ and
‘tollgates’ in the route which has been selected by the ant ‘k.’ Taking into
account the complexities involved in computing convolutions, only two input
fuzzy sets, “low” and “high” are defined for each input. The architecture of
fuzzy system is in Figure 5.3.

![Figure 5.3 Architecture of the fuzzy logic system](image)

The fuzzy rule base is in Table 5.1.
<table>
<thead>
<tr>
<th>Rule no.</th>
<th>IF</th>
<th>THEN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quality</td>
<td>Road condition</td>
</tr>
<tr>
<td>1</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>2</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>3</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>5</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>6</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>7</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>8</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>9</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>10</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>11</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>12</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>13</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>14</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>15</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>16</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

There are 16 rules defined for this fuzzy system. The membership functions for the fuzzy input are in Figure 5.4.
The universe of discourse for the fuzzy variables are all normalized between (0,1). The membership functions for the fuzzy sets of inputs are chosen to be trapezoidal-shaped because this type of membership function has good features such as easiness in computation.

There is also a fuzzy set for output variable in Figure 5.5.
All the membership functions for the fuzzy sets of inputs are chosen to be trapezoidal in shape since such a shape with its fine features easily yields to uncomplicated computation.

The output variable has four membership functions titled, very strong, very weak, strong and weak. The fuzzy operator used for the AND method is “if-then rules” such as “IF a is a1 and b is b1, then c is c1”. The defuzzification is the process of conversion of fuzzy output set into a single number.

5.8.1 Ant Based Routing

The fuzzy ant based routing algorithm is in Figure 5.6.

![Procedure FLACS](image)

Figure 5.6 Fuzzy ant based routing algorithm

- **Initialize.** It is the initialization of the algorithm parameters such as number of ants and evaporation coefficient.
- **Locate ants.** Ants are located at the point of origin in this stage. It is also verified whether ant is free or blocked. Each ant can traverse each junction once in each iteration.

- **Construct Probability:** The probability of each possible direct route is calculated based on the total number of edges for each active ant. The probability of displacing from junction $i$ to junction $j$ for ant $k$ is

$$
P_{ij}^k = \begin{cases} 
\frac{\tau_{ij} \prod_{l \in \text{parameters}} \xi_{ih_l}^{-\gamma h_l}}{\sum_{j \notin \text{tabu}_k} \tau_{ij} \prod_{l \in \text{parameters}} \xi_{ih_l}^{-\gamma h_l}} & \text{if } j \notin \text{tabu}_k, \\
0 & \text{otherwise}
\end{cases}
$$

where $\tau_{ij}$ is the direct route pheromone intensity from junction $i$ to $j$. Parameters $\alpha$ controls the importance of $\tau_{ij}$. The $\text{tabu}_k$ list is the set of direct blocked routes. Parameter set is a collection of most important parameters for drivers taking journeys in metropolises. To make it the simplest, the parameters such as traffic, road condition, quality and tollgates are considered in this set. Cost function of each parameter $l$ is adjustable by $\alpha$. If the traffic is heavy and the tollgates are many, the total cost increases and consequently there is less probability of selecting that route. If the road condition is good and the volume of traffic is low, it reduces the total cost and increases the probability of selecting that route.

- **Select route.** The route with the highest probability is always selected by the active ant. Otherwise, the next better junction is aimed at through probabilities.
- **Update Tabu List.** In this step, the route chosen by ant $k$ is added to the tabu list in order not to avoid it in future selection.

- **Update Pheromone.** The Ant Pheromone System comprises of two main rules. First one is the Applied Local Pheromone Update rule Second one is the Global Pheromone Update rule which is put into operation after all ants have finished constructing a solution.

The pheromone amount of the route junction $i$ and $j$ is updated for ant $k$ with the following equation

$$
\tau_{ij}^{new} = \tau_{ij}^{old} + (10X\lambda \tau)
$$

(5.7)

where $\lambda \tau$ is the amount of local pheromone updating. The value of $\lambda \tau$ is the output of a fuzzy logic system.

The total last step of each completed loop is global pheromone updating defined as

$$
\tau_{ij}^{new} = \rho \tau_{ij}^{old}
$$

(5.8)

where $0 < \rho < 1$ is the evaporation coefficient and is usually set to 0.9.

- **Select Best Direction.** After $m$ loops, the direction which offers the lowest cost from origin to destination is recommended by the system.
5.9 EXPERIMENTAL EVALUATION

The theoretical representation of hierarchical routing algorithm has to be assessed through experimental verification. A simulated road network with 25 vertices and 34 edges is assumed. All algorithms are developed in C# .net framework and conducted on Intel ® Core ™ i3-2120 CPU @3.30 GHz. The OS of the system is Microsoft Windows.

5.9.1 Preprocessing

The community detection algorithm is operated with a variation of edge weight in the number of vertices divided by the distance of the road. Such close intersections are more likely to form the same community. This scheme is applied on road network. The network consists of 25 vertices and 34 edges.

Figure 5.7 shows the primary input of creating the graph and exporting the excel file with existing source, target and weight.

![Figure 5.7 Original graph](image-url)
It shows the secondary input from importing the excel file using exported file to avoid recreation of the graph. Figure 5.8 shows the community graph with various color vertices.

Figure 5.8 Community Formation

Figure 5.9 shows the high-level community graph with 6 communities and 15 border nodes.

Figure 5.9 High level community
The hierarchical approach is made use of to limit the storage space and computational cost. The parameter values about the edges for all selected border nodes are obtained from the user. The inputs may be low or high for the user’s parameters, traffic, tollgate, quality, and road condition. The illustration is explicated in Table 5.2. By studying the table, we can conclude naturally that the ants would choose the route that is very strong in terms of dynamic results.

Table 5.2 Local pheromone updating

<table>
<thead>
<tr>
<th>No of border node edges</th>
<th>Border edge</th>
<th>User preference</th>
<th>Dynamic Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Quality</td>
<td>Road condition</td>
</tr>
<tr>
<td>1.</td>
<td>7-14</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>2.</td>
<td>7-13</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

5.10 DYNAMIC SHORTEST PATH

In order to arrive at the shortest path, the first step involves the selection of the source and destination pair randomly and also deciding upon the users’ parameters.

5.10.1 Intra Community Result

If it is imagined that the source and destination pair opted for is from the same community, then Dijkstra’s algorithm could be relied upon for achieving the shortest path. In the current representation, the range is 0/7. The results arrived at find figurative expression in Figure 5.10. The 0-2-4-7 path is highlighted with green color.
5.10.2 Inter Community Result

Let another conjecture be analyzed. The source and destination are taken from two separate communities. The inputs are imagined as follows.

- The edges are 7-24
- Traffic is 20%
- Tollgate is 10%
- Quality is 90%
- Road condition is 80%

For these parameters, the local pheromone value obtained is 84.7% and this value, in terms of dynamic result analysis, should be ‘very strong.’ The investigation is presented in Figure 5.11.
The fuzzy ant based algorithm updates the local pheromone and the dynamic result obtained is ‘very strong.’ The preference of 7/24 vertices is made with regard to source and destination by bringing into play the Fuzzy Ant-based algorithm.

Another probable routing pattern with a number of edges is expressed in Table 5.3.

**Table 5.3  Efficient dynamic shortest path routing**

<table>
<thead>
<tr>
<th>Route selection system</th>
<th>s/t pair</th>
<th>Possible Routing</th>
<th>Number of edges with dynamic result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy-Ant</td>
<td>7/24</td>
<td>7-14-16-23-22-24</td>
<td>6 with Very strong</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7-13-21-19-24</td>
<td>5 with Strong</td>
</tr>
</tbody>
</table>
While in the possible routing pattern, 7-14-16-23-22-24, the edge weight is 6, the edge weight in the other routing, 7-13-21-19-24, is five. There are two ways from the source, 7-14 and 7-13. From these two edges, the stronger local pheromone is updated by the ant. Therefore, it is apparent that the ant has made the choice, 7-14, from multiple alternatives.

Whereas Figure 5.11 puts on show the efficient dynamic routing in tabular form, the graphical organization of the same with the application of hierarchical fuzzy ant algorithm gets highlighted in Figure 5.12.

**Figure 5.12 Inter community dynamic shortest path routing**

### 5.11 CONCLUSION

This chapter is, thus a study of the problem of route selection in large road networks with a specific dimension. The process involves the application of several efficient algorithms – hierarchical community structure, fuzzy logic and ant colony system. There already exists numerous systems in the configuration of routing and identifying the shortest path in large road
networks with both static and dynamic state of affairs but they could not be easily implemented what with their mind-boggling storage as well as computational cost.

This chapter is an endeavour to develop a hierarchical approach that becomes an effective tool in efficient route computation on large road networks. The hierarchical community mining algorithm has been of assistance in scheming a hierarchical graph model, which could compute optimal dynamic routes for same community nodes pair and different community nodes pair on large road networks, based on user parameters of traffic, tollgate, quality and road condition by using fuzzy logic and ant colony system.

Fuzzy logic is regarded as a management mechanism for the exposition of ant colony system local pheromone updating. The experimental results demonstrate that the algorithm encompasses lots of real–time, wide-ranging and all-purpose applications for emergency services, tourism and to be precise, universally, for anyone who wants to have a low-cost, safe and comfortable journey in large road networks.

The succeeding chapter dwells upon the design of user based decision support system in route selection on road network. Dijkstra’s shortest path algorithm rooted in dynamic approach is taken up as the base on which the structure of the system is developed.