Fuzzy Shortest Route Algorithm for Telephone Line Connection Using Matrix Algorithm

R. Kalaivani¹, Dr. D. Vijayalakshmi²

¹Research Scholar, Department of Mathematics, Kongunadu Arts and Science College, Coimbatore-641 029, India
²Assistant Professor, Department of Mathematics, Kongunadu Arts and Science College, Coimbatore-641 029, India

Abstract: In computer science, there are many algorithms that finds a minimum spanning tree for a connected weighted undirected fuzzy graph. The minimum length (or cost) spanning tree problem is one of the nicest and simplest problems in network optimization, and it has a wide variety of applications. The problem is to find a minimum cost (or length) spanning tree in G. Applications include the design of various types of distribution networks in which the nodes represent cities, centers etc.; and edges represent communication links (fiber glass phone lines, data transmission lines, cable TV lines, etc.), high voltage power transmission lines, natural gas or crude oil pipelines, water pipelines, highways, etc. The objective is to design a network that connects all the nodes using the minimum length of cable or pipe or other resource. In this paper we find the solution to the problem is to minimize the amount of new telephone line connection using matrix algorithm with fuzzy graph.

Keywords: spanning tree – Nearest Neighbor Algorithm– Minimum weight link Algorithm– Kruskal’s Algorithm– Matrix algorithm.

1. Introduction

A minimum cost of the fuzzy spanning tree is spanning tree, but it has weight or length associated with the edges and total weight of the tree is minimum. A fuzzy tree for that graph would be a subset of those paths that has no cycles but still connects to every vertex. There might be several spanning trees possible. A minimum fuzzy tree would be one with the lowest total cost. A less obvious application is that the minimum fuzzy spanning tree can be used to approximately solve the traveling salesman problem.

In this paper, we find the solution for the problem, [1] that (A.V.V.M. Sri Pushpam College, (Autonomous), Poondi, Thanjavur District, South India) needs to connect updated intercom lines connecting all the departments using a fuzzy shortest route algorithm. The problem is to minimize the amount of new line using matrix Algorithm with fuzzy graph.

2. Definition

Definition 2.1:

A fuzzy graph with V as the underlying set is a pair G: (A, Γ) where A: V → [0,1] is a fuzzy subset, Γ : VxV ----> [0,1] is a fuzzy relation on the fuzzy subset A, such that Γ(u,v) ≤ A(u) ∩ A(v) for all u,v ∈ V.

Definition 2.2:

A fuzzy Hamiltonian circuit is a circuit that visits every vertex in a fuzzy graph once with no repeats, being fuzzy Hamiltonian circuits must start and end at the same vertex.

Definition 2.3:

A fuzzy Hamiltonian path is a path that passes through each of the vertices in a fuzzy graph exactly once.

Definition 2.4:

A fuzzy spanning tree is a fuzzy tree which covers all the vertices of a fuzzy graph.

NOTE 2.1: Fuzzy trees has no circuits, and it is fine to have vertices

2.1 Sorted Edges Algorithm (Minimum Weight Link Algorithm)

Step 1: Select the minimum weighted unused edge in the fuzzy graph; highlight it.

Step 2: Repeat step 1, adding the cheapest unused edge to the fuzzy graph, unless:

a) Addition of edge would create a circuit that doesn’t contain all vertices, or
b) Addition of edge would give a vertex degree 3.

Step 3: Repeat until a fuzzy Hamiltonian circuit containing all vertices is formed.

Example: 2.1, we have five vertex fuzzy graphs; the minimum weight edge is x₁x₅, with a weight of 0.02. The next fuzzy shortest edge is x₅x₄, with a weight of 0.05. The next fuzzy shortest edge is x₄x₃, with a weight of 0.12. The next fuzzy shortest edge is x₃x₂, with a weight of 0.07. The next fuzzy shortest edge is x₂x₄, but that edge would create a fuzzy Hamiltonian circuit x₁x₃x₄x₂x₄.
Example 2.2:

The telephone line connection to plan and connect an efficient fuzzy route for [7] for intercom landline to reach it all the department exactly once and return to starting location. (eg. Office room). Consider each department as vertex such as: x₁ – office room, x₂ – mathematics, x₃ – economics, x₄ – History, x₅ – Computer Science, x₆ – Library, x₇ – Physics, x₈ – Chemistry, x₉ – Botany and x₁₀ – Physical education.

The distance between them are represented as fuzzy weights in the tabular column 3.1.

<table>
<thead>
<tr>
<th></th>
<th>x₁</th>
<th>x₂</th>
<th>x₃</th>
<th>x₄</th>
<th>x₅</th>
<th>x₆</th>
<th>x₇</th>
<th>x₈</th>
<th>x₉</th>
<th>x₁₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>x₁</td>
<td>-</td>
<td>0.37</td>
<td>0.1</td>
<td>0.22</td>
<td>0.07</td>
<td>0.17</td>
<td>0.25</td>
<td>0.28</td>
<td>0.24</td>
<td>0.35</td>
</tr>
<tr>
<td>x₂</td>
<td>0.37</td>
<td>-</td>
<td>0.25</td>
<td>0.16</td>
<td>0.5</td>
<td>0.19</td>
<td>0.07</td>
<td>0.1</td>
<td>0.13</td>
<td>0.01</td>
</tr>
<tr>
<td>x₃</td>
<td>0.1</td>
<td>0.25</td>
<td>-</td>
<td>0.12</td>
<td>0.27</td>
<td>0.12</td>
<td>0.09</td>
<td>0.16</td>
<td>0.13</td>
<td>0.24</td>
</tr>
<tr>
<td>x₄</td>
<td>0.22</td>
<td>0.16</td>
<td>0.12</td>
<td>-</td>
<td>0.43</td>
<td>0.03</td>
<td>0.04</td>
<td>0.08</td>
<td>0.02</td>
<td>0.15</td>
</tr>
<tr>
<td>x₅</td>
<td>0.07</td>
<td>0.5</td>
<td>0.27</td>
<td>0.43</td>
<td>-</td>
<td>0.45</td>
<td>0.47</td>
<td>0.34</td>
<td>0.38</td>
<td>0.42</td>
</tr>
<tr>
<td>x₆</td>
<td>0.17</td>
<td>0.19</td>
<td>0.12</td>
<td>0.03</td>
<td>0.45</td>
<td>-</td>
<td>0.06</td>
<td>0.11</td>
<td>0.06</td>
<td>0.18</td>
</tr>
<tr>
<td>x₇</td>
<td>0.25</td>
<td>0.07</td>
<td>0.09</td>
<td>0.04</td>
<td>0.47</td>
<td>0.06</td>
<td>-</td>
<td>0.11</td>
<td>0.08</td>
<td>0.11</td>
</tr>
<tr>
<td>x₈</td>
<td>0.28</td>
<td>0.1</td>
<td>0.16</td>
<td>0.08</td>
<td>0.34</td>
<td>0.11</td>
<td>0.11</td>
<td>-</td>
<td>0.03</td>
<td>0.05</td>
</tr>
<tr>
<td>x₉</td>
<td>0.24</td>
<td>0.13</td>
<td>0.13</td>
<td>0.02</td>
<td>0.38</td>
<td>0.06</td>
<td>0.08</td>
<td>0.03</td>
<td>-</td>
<td>0.11</td>
</tr>
<tr>
<td>x₁₀</td>
<td>0.35</td>
<td>0.01</td>
<td>0.24</td>
<td>0.15</td>
<td>0.42</td>
<td>0.18</td>
<td>0.11</td>
<td>0.05</td>
<td>0.11</td>
<td>-</td>
</tr>
</tbody>
</table>
2.2 Nearest Neighbour Algorithm (NNA)

Step 1: 
Select a starting point.

Step 2: 
Move to the nearest unvisited vertex (the edge with smallest weight).

Step 3: 
Repeat until the circuit is complete.

Using NNA starting at $x_8$ (Nearest Neighbor Algorithm) with a large number of departments, it is helpful to mark off the departments as they are connected to keep from accidentally connecting them again. Looking in the row for $x_8$, the smallest distance is 0.03, to $x_9$. Following that idea, our circuit will be:

<table>
<thead>
<tr>
<th>Table 2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_8$ To $x_9$ 0.03</td>
</tr>
<tr>
<td>$x_9$ To $x_4$ 0.02</td>
</tr>
<tr>
<td>$x_4$ To $x_6$ 0.03</td>
</tr>
<tr>
<td>$x_6$ To $x_7$ 0.06</td>
</tr>
<tr>
<td>$x_7$ To $x_{10}$ 0.11</td>
</tr>
<tr>
<td>$x_{10}$ To $x_2$ 0.01</td>
</tr>
<tr>
<td>$x_2$ To $x_3$ 0.25</td>
</tr>
<tr>
<td>$x_3$ To $x_1$ 0.1</td>
</tr>
<tr>
<td>$x_1$ To $x_5$ 0.07</td>
</tr>
<tr>
<td>$x_5$ To $x_8$ 0.34</td>
</tr>
</tbody>
</table>

Therefore, total connecting length is 1 (approximately).

Using **SORTED EDGES ALGORITHM** it is helpful to draw an empty fuzzy graph, perhaps by drawing vertices in a circular pattern. Adding edges to the fuzzy graph and select them with help of any fuzzy circuits or vertices with degree 3.

The fuzzy graph after adding these edges is shown in the following figure 3.3.

The next shortest edge is from $x_4$ to $x_7$ at 0.04, but adding that edge would give $x_4$ degree 3. Continuing on, and skip over any edge pair that contains $x_9$ to $x_4$, since they both already have degree 2.

- $x_4$ to $x_{10}$ 0.05
- $x_9$ to $x_7$ 0.06
- $x_3$ to $x_2$ (reject – it forms fuzzy circuit shown in fig. 3.4)
- $x_1$ to $x_5$ 0.07

The fuzzy graph after adding these edges is shown in the following figure 3.4. At this point, skip over any edge pair that contains $x_9$, $x_{10}$, $x_6$, $x_8$ or $x_4$ since they already have degree 2.

- $x_7$ to $x_2$ (reject – it forms fuzzy circuit)
- $x_7$ to $x_3$ 0.09
- $x_3$ to $x_1$ 0.1

The fuzzy graph after adding these edges is shown in the following figure 3.5.
At this point the only way to complete the circuit is to add:
\[ x_3 \text{ to } x_2 = 0.5 \]

The total length of the cable
\[ = [(x_{10}, x_2) + (x_3, x_9) + (x_8, x_9) + (x_6, x_9) + (x_1, x_3) + (x_7, x_3) + (x_5, x_2)] 
= 0.01 + 0.02 + 0.03 + 0.03 + 0.06 + 0.07 + 0.09 + 0.1 + 0.5 
= 0.96 \]

2.3 Kruskal’s Algorithm

Step 1:
Select the minimum weight unused edge in the fuzzy graph; highlight it.

Step 2:
Repeat step 1, addition of minimum weight unused edge to the fuzzy graph, unless: Addition of edge would create a fuzzy circuit.

Step 3:
Repeat until a fuzzy spanning tree is formed.

Example: 3.4
The telephone line connection to plan and connect an efficient fuzzy route for [1]. Consider each department as vertex such as \( x_1 \) – office room, \( x_2 \) – mathematics, \( x_3 \) – economics, \( x_4 \) – History, \( x_5 \) – Computer Science, \( x_6 \) – Library, \( x_7 \) – Physics, \( x_8 \) – Chemistry, \( x_9 \) – Botany and \( x_{10} \) – Physical education. The distances between them are represented as fuzzy weights shown in the Tabular column 3.1.

Solution:
Selecting the cheapest unused edge in the fuzzy graph, we have the following results.

<table>
<thead>
<tr>
<th>Edge</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_{10} ) to ( x_2 )</td>
<td>0.01</td>
</tr>
<tr>
<td>( x_4 ) to ( x_9 )</td>
<td>0.02</td>
</tr>
<tr>
<td>( x_8 ) to ( x_9 )</td>
<td>0.03</td>
</tr>
<tr>
<td>( x_4 ) to ( x_4 )</td>
<td>0.03</td>
</tr>
<tr>
<td>( x_4 ) to ( x_7 )</td>
<td>0.04</td>
</tr>
<tr>
<td>( x_9 ) to ( x_6 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_8 ) to ( x_{10} )</td>
<td>0.05</td>
</tr>
<tr>
<td>( x_7 ) to ( x_9 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_6 ) to ( x_7 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_8 ) to ( x_2 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_1 ) to ( x_3 )</td>
<td>0.07</td>
</tr>
<tr>
<td>( x_6 ) to ( x_8 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_7 ) to ( x_7 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_7 ) to ( x_{10} )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_9 ) to ( x_{10} )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_3 ) to ( x_4 )</td>
<td>0.12</td>
</tr>
<tr>
<td>( x_3 ) to ( x_9 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_2 ) to ( x_7 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_9 ) to ( x_2 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_4 ) to ( x_{10} )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_9 ) to ( x_7 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_6 ) to ( x_3 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_8 ) to ( x_4 )</td>
<td>(reject – it forms fuzzy circuit)</td>
</tr>
<tr>
<td>( x_6 ) to ( x_1 )</td>
<td>0.17</td>
</tr>
</tbody>
</table>
Accepted path is connected and shown in the following figure 3.6.

![Figure 2.6: Updated telephone line connection](image)

This connects the fuzzy graph.

The total length of cable lay would be

\[
= \{(x_{10},x_2)+(x_4,x_9)+(x_6,x_8)+(x_4,x_7)+(x_8,x_9),(x_1,x_3)\}+(x_7,x_5)+(x_6,x_1)\\
= 0.01+0.02+0.03+0.03+0.04+0.05+0.07+0.12+0.17\\
= 0.54
\]

### 2.4 Matrix Algorithm

**Input:** the weight matrix \( M = [w_{ij}]_{n \times n} \) for the undirected weighted graph \( G \)

**Step 1:**
- Start

**Step 2:**
- Repeat Step 3 to Step 4 until all (n-1) elements matrix of \( M \) are marked or set to zero or in other words all the nonzero elements are marked
- Search the weight matrix \( M \) by row-wise to find the unmarked nonzero minimum element \( M_{[i,j]} \), which is the weight of the corresponding edge \( e_{ij} \) in \( M \).
- If the corresponding edge \( e_{ij} \) of selected \( M_{[i,j]} \) forms cycle with the already marked elements in the elements of the \( M \) then Set \( M_{[i,j]} = 0 \) else mark \( M_{[i,j]} \)
- Construct the graph \( T \) including only the marked elements from the weight matrix \( M \) which shall be the desired Minimum cost spanning tree of \( G \).

**Output:**
- Minimum Cost Spanning Tree \( T \) of \( G \).

**Example:**

In this example we applied the matrix algorithm to plan and connect an efficient fuzzy route telephone line connection for the problem [7]. Consider each department as vertex such as \( x_1 \) – office room, \( x_2 \) – mathematics, \( x_3 \) – economics, \( x_4 \) – History, \( x_5 \) – Computer Science, \( x_6 \) – Library, \( x_7 \) – Physics, \( x_8 \) – Chemistry, \( x_9 \) – Botany and \( x_{10} \) – Physical education. The distance between them are represented as fuzzy weights matrix shown in the fig.3.8.

In this section, we find the solution for the below matrix the table is taken from [7] needs to connect updated intercom lines connecting all the departments exactly once. The problem is to minimize the amount of new line using Matrix Algorithm [8] with fuzzy graph. The step by step procedure is given below.

**Distance between the departments is shown in below matrix:**

\[
\begin{array}{cccccccccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\
 x_1 & - & 0.37 & 0.04 & 0.22 & 0.07 & 0.17 & 0.25 & 0.28 & 0.24 & 0.35 \\
x_2 & 0.37 & - & 0.25 & 0.16 & 0.5 & 0.19 & 0.07 & 0.1 & 0.13 & 0.01 \\
x_3 & 0.1 & 0.25 & - & 0.12 & 0.27 & 0.12 & 0.02 & 0.16 & 0.13 & 0.24 \\
x_4 & 0.22 & 0.16 & 0.12 & - & 0.43 & 0.03 & 0.04 & 0.04 & 0.02 & 0.15 \\
x_5 & 0.07 & 0.5 & 0.27 & 0.43 & - & 0.45 & 0.47 & 0.34 & 0.38 & 0.42 \\
x_6 & 0.17 & 0.19 & 0.12 & 0.04 & 0.45 & - & 0.06 & 0.11 & 0.06 & 0.18 \\
x_7 & 0.25 & 0.07 & 0.09 & 0.04 & 0.47 & 0.00 & - & 0.11 & 0.08 & 0.11 \\
x_8 & 0.28 & 0.1 & 0.16 & 0.08 & 0.34 & 0.11 & 0.11 & - & 0.03 & 0.05 \\
x_9 & 0.24 & 0.13 & 0.13 & 0.02 & 0.38 & 0.06 & 0.08 & 0.03 & - & 0.11 \\
x_{10} & 0.35 & 0.01 & 0.24 & 0.15 & 0.42 & 0.18 & 0.11 & 0.05 & 0.11 & - \\
\end{array}
\]

**Figure 2.7: Weight matrix**
From the above fig 3.7 the minimum weight is selected and colored, their corresponding edges were drawn repeat the process until the algorithm terminates. While drawing the graph if its forms fuzzy circuit, we remove that and in the table the weight is marked as 0.

<table>
<thead>
<tr>
<th>x_2</th>
<th>to</th>
<th>x_{10}</th>
<th>0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_4</td>
<td>to</td>
<td>x_6</td>
<td>0.02</td>
</tr>
<tr>
<td>x_4</td>
<td>to</td>
<td>x_7</td>
<td>0.03</td>
</tr>
<tr>
<td>x_4</td>
<td>to</td>
<td>x_8</td>
<td>0.04</td>
</tr>
<tr>
<td>x_8</td>
<td>to</td>
<td>x_{10}</td>
<td>0.05</td>
</tr>
<tr>
<td>x_7</td>
<td>to</td>
<td>x_6</td>
<td>(forms fuzzy circuit) its shown in below Fig.3.7</td>
</tr>
<tr>
<td>x_1</td>
<td>to</td>
<td>x_5</td>
<td>0.07</td>
</tr>
<tr>
<td>x_4</td>
<td>to</td>
<td>x_8</td>
<td>0.08</td>
</tr>
<tr>
<td>x_3</td>
<td>to</td>
<td>x_7</td>
<td>0.09</td>
</tr>
<tr>
<td>x_1</td>
<td>to</td>
<td>x_3</td>
<td>0.1</td>
</tr>
</tbody>
</table>

**Fig 2.8: Updated fuzzy route telephone line connection**

Using the Matrix Algorithm

The length of the cable

\[
\{(x_2,x_{10})+(x_4,x_6)+(x_4,x_7)+(x_4,x_9)+(x_6,x_{10})+(x_1,x_3)+(x_1,x_7)+(x_4,x_8)\} = 0.01+0.02+0.03+0.04+0.05+0.07+0.08+0.09+0.1 = 0.49.
\]

### 3. Conclusion

In this paper we of comparing with heuristic algorithm such as the Nearest Neighbour Algorithm, Sorted Edges Algorithm, Kruskal’s Algorithm and Matrix Algorithm for the problem that connecting distances to plan and visit an efficient fuzzy spanning tree route for the intercom land line (phone) to reach it all the department exactly once. So we conclude that Matrix Algorithm is the best to adopt for these types of problems.

### References


[21] NarsinghDeo, “Graph Theory with Applications to Engineering and Computer Science”, PHI Learning, 2011.


