Jurnal Teknologi Informasi dan Komunikasi

Vol: 15 No 02 2024 E-ISSN: 2477-3255

Accepted by the Editor: 10-09-2024 | Revision: 16-11-2024 | Published: 30-11-2024

Solution Approach to the Minimum Spanning Tree Problem in Tsukamoto Fuzzy and Fermantean Fuzzy Environments

Citra Fathia Palembang [1], Desli Inggrit de Haas[2], Abd. Aziz Rumakefing [3]

[1,2,3]Program Studi Ilmu Komputer Universitas Pattimura

[1,2,3]Jl. Ir. M. Putuhena, Ambon, Maluku, Indonesia

e-mail: [1]fpchiet@gmail.com, [2]dehaasdesli@gmail.com, [3]Abdulazizrumakeffing01@gmail.com

Abstract

Solving Fuzzy Minimum Spanning Tree (FFMST) and Fuzzy Tsukamoto using modified Prim Algorithm for Undirected Graphs and modified Optimal Branching Algorithm for Directed Graphs in FFN environment. Since the proposed Algorithm includes FFN ranking and Arithmetic Operations, we use the improved FFN scoring function to compare the edge weights of the graphs. With the help of Numerical examples, the solution technique for the proposed FFMST model is explained.It aims to modify the Prims algorithm for oriktade graphing and the optimal result processing algorithm for re-graphing in Fuzzy Fermatean (FFN)-miljö. They utilize the finite FFN function and the operation of fuzzy function operations to ensure victory in graphing. The fuzzy-inference process is based on the Tsukamoto method and also to get the best result from the existing catch.Numerical examples of presenters to perform the tasks of missing presenters. The results are seen in the effective Prim algorithm modifier lost Fuzzy Fermatean MST -problem for genome oriktade generator generated at minimum cost and fall with local banks. It is an optimal business growth modifier to optimize services for lenders, such as communication between financial consultants and commercial banks. This method will be effective and increase the desired parameters. Tsukamoto Fuzzy -This method includes a fuzzy-inference process to get the best answer in the minimum spanning tree problem. Kantvikter functions based on levels of capability and range functions. Theminimum spanning tree is achieved by the Prims algorithm, which may be performed with fuzzy values first.

Keywords: *Fuzzy, Tsukamoto, Fermatean, Algorithm, Graph, Minimum Spanning Tree, Fuzzy, Fuzzy Set Theory, Prim Algorithm.*

1. Introduction.

In conventional graph theory, the minimal spanning tree (MST) [1] is a commonly used combinatorial optimization problem. Many researchers love it. Czech scientist Otakar Borůvka in 1926 was the first to propose an algorithm in finding the MST. Later versions of the algorithm were by Prim [2] and Dijkstra in 1959 and further developed by Kruskal [3]. Therefore, many researchers have diligently studied many things to create an effective MST algorithm. Graham and Hell [4] explored the central role in the history of MST. The importance

https://doi.org/10.31849/digitalzone.v15i2. 22826

and popularity of MST is due to its real-world applications such as designing network problems in transportation, telecommunications, and water supply.

MST edge weights can be calculated using arbitrary values assigned to edges, including distance, traffic load, overcrowding, and so on. Uncertainty is not properly represented in classical graph theory and therefore almost all MST problems have weights assigned to the edges as real numbers, but in relevant problems in daily life these parameters are naturally imprecise, leading to uncertainty and ambiguity. To overcome the ambiguity and uncertainty, some researchers have used random variables to overcome the lack of precision of the arc weights. This particular type of MST problem is characterized as a stochastic minimum spanning tree (SMST) problem [5]-[7].

This assumption may not be true under realistic circumstances in stochastic MST problems. This can be done when the probability distribution functions of the edge weights are assumed to be known, which is the main problem of the SMST algorithm. Fuzzy set theory (FS) proposed by Zadeh [8] is a flexible and efficient tool for handling transactions with imprecise and vague information. To articulate imprecise information precisely, Atanassov [9] proposed intuitionistic fuzzy sets (IFS) as an extension of FS by combining the degree of membership (m) with the degree of non-membership (n) of each element such that $0 \leq m + n \leq 1$.

However, if $m + n > 1$, then IFS no longer applies. To overcome this Yager [10], [11] proposed the pythogorean fuzzy set (PFS) with the condition $0 \leq m2 + n\ 2 \leq 1$. Although PFS generalizes IFS, it cannot define uncertainty when the condition fails and hence Senapati and Yager [12], proposed the fermatean fuzzy set (FFS) by enlarging the uncertainty domain with the condition $0 \leq m3 + n\ 3 \leq 1$ and explored the properties of FFS. Rosenfeld [13] proposed fuzzy analogs of various conceptions of basic graph theory and Bhattacharya [14], gave a note on fuzzy graphs (FG). Shannon and Atanassov [15], introduced the concepts of intuitionistic fuzzy relations (IFR) and intuitionistic fuzzy graphs (IFG).

Implementation of Fuzzy Logic to determine the amount of money spent in the bank money withdrawn in large or small amounts for daily needs, fuzzy logic is used to make it easier for customers to find out the amount of money spent each month Based on the data obtained from the bank, by utilizing the fuzzification process, fuzzy inference and fuzzy rule base, and defuzzification in fuzzy logic, fuzzy logic can be applied to determine the amount of money spent. From the research results, there are nine membership functions from two input parameters, namely money withdrawal and remaining balance. The linguistic variables used are Little Expenditure, Medium Expenditure, Much Expenditure, Little Balance, Medium Balance, and, Much Balance [16]. The following are the reasons why we chose the fermantean fuzzy approach compared to other fuzzy approaches:

a. Fuzzy Fermatean can express the degree of membership (degree of membership), the degree of non-membership), and the degree of doubt) of type-1 fuzzy sets or even intuitionistic fuzzy sets.
b. In MST, relationships between nodes often contain complex information such as cost, time, or risk. Fermatean fuzzy allows decision making by considering these complex relationships more realistically compared to Tsukamoto fuzzy which tends to be simpler.
c. Fuzzy fermateans offer a wider parameter space, which is useful in modeling MST parameters such as distance or cost. In problems like MST, more flexible space parameters often lead to more optimal solutions.
d. The fuzzy Fermatean approach is more resistant to incomplete or noisy data, which is often encountered in attribute measurements for MST problems. In contrast, approaches such as Tsukamoto fuzzy have higher sensitivity to noise because they rely on rigid inference rules.

Fuzzy fermatean solves MST by representing each relationship attribute in the form of degree of membership, degree of non-membership, degree of doubt The process involves, Fuzzification with customized collection functions, Determination of weights based on Fermatean fuzzy aggregation, MST algorithms such as Prim or Kruskal are modified to work with fuzzy Fermatean values. This approach allows for more accurate results and is robust to ambiguous data.

The comparative approaches to extend FMST in Table 2 focus on the suitability of the research and reinforcement for this article. A thorough review of the previous FMST literature reveals that IFS and PFS environments are the only topics discussed. Although previous studies have successfully applied the MST concept in fuzzy environments, there are still limitations in handling the complexity of networks involving fuzzy membership functions (FFNs). Conventional MST algorithms, such as Prim and Kruskal's algorithm, are designed for graphs with fixed weights, while in fuzzy environments, edge weights are expressed in terms of FFNs, which require different computational approaches. The following are the main contributions of the proposed work :

A. We have proposed an algorithmic approach to the FFMST problem for both directed and undirected FFGs.
B. For directed FFGs, an innovative concept of modified optimal branching algorithm is proposed for the first time in FFN environment.
C. For undirected FFG, a modified Prim algorithm is proposed in the FFN environment. The existing FFN score function and arithmetic operations are utilized in the proposed algorithm.
D. To prove the efficiency of the suggested modified Prim algorithm, we have considered the problem of a local bank that wants to build its network connecting headquarters, branches and ATMs as an example for undirected FFG and obtain the lowest possible cost by using the proposed Prim algorithm.
E. In addition, we have discussed another example of directed graph to explore the effectiveness of the modified optimal branching algorithm.

The rest of the paper is organized as follows. We examine the initial data in section 2, with respect to FFG, score function and FFN arithmetic operations. A new algorithm is proposed in section 3, based on the traditional Prim algorithm and a modified optimal branching algorithm is developed to determine FMST. In section 4, illustrative examples are discussed. Finally, the conclusion and future work of the proposed algorithm are discussed in section 5. The purpose of this article is to bridge this gap by incorporating MST into the context of FFN. The main contribution of our proposed research is to solve the FFMST whose edge weights are FFNs by using a modified Prim algorithm for undirected graphs and a modified optimal branching algorithm for directed graphs. Thus, this research makes a significant contribution in extending the applicability of MST into more complex and realistic domains.

2. Research Methods.

A. Tsukamoto Fuzzy Method

The steps taken in this research can be seen in the flowchart in Figure

B. Modifying Prim's Algorithm for Undirected FFMST.

Consider an undirected FFG. To minimize the total edge weight, the modified Prim's algorithm identifies a subset of the edge set $\mathbb{B}$ that contains all the vertices of the FFG. The following are the steps involved in the modification of Prim's algorithm. A. Input: Undirected FFG. B.
Step 1: select an arbitrary starting point from the vertex set $\mathbb{A}$ and declare it as the root point. C.
Step 2: select the edge connecting the tree vertex and the fringe vertex that has the minimum edge weight. The minimum edge weight of FFMST can be calculated using SF FFN. D.
Step 3: add the selected edge to the FFMST if it does not form a closed cycle. e.
Step 4: repeat steps 3 and 4 until the fringe vertices (vertices not included in the FFMST) remain. F. Output: $\mathscr{F}$, the FFMST for the given undirected graph.

C. Modifying the Optimal Branching Algorithm for FFMST Directed Graphs

Consider a directed FFG. To minimize the total edge weight, the modified optimal branching algorithm identifies a subset of the edge set $\mathbb{B}$ that contains all the vertices of the FFG. The

https://doi.org/10.31849/digitalzone.v15i2. 22826

following are the steps involved in the modified optimal branching algorithm. A. Input: A directed FFG rooted at $r \in \mathbb{A}$ b.
Step 1: determine the score function for the FFN. C.
Step 2: using the FFN, create FFGs with their respective edge weights.
It is important to ensure that the edge weights obtained are not negative. D.
Step 3: initialize the root $r \in \mathbb{A}$ as the source. e.
Step 4: add the following vertices now, in any order. F.
Step 5: analyze the degree of the selected vertex and find the minimum weighted edge entering the selected vertex. Rewrite its weight by using the determined minimum weight difference. G.
Step 6: repeat step 5 for all $v \neq a \in \mathbb{A}$. H.
Step 7: construct a subgraph $\mathscr{F}$ of $\mathcal{G}\mathcal{J}$ rooted at r such that $\mathscr{F}$ has no directed cycles and every node $v \neq a$ has exactly one incoming edge where the required spanning tree $\mathscr{F}$ is located. I.
Step 8: given all possible graphs of $\mathscr{F}$, compute the arborescence rooted at r with minimum cost.
Output: $\mathscr{F}$, FFMST for a given directed graph.

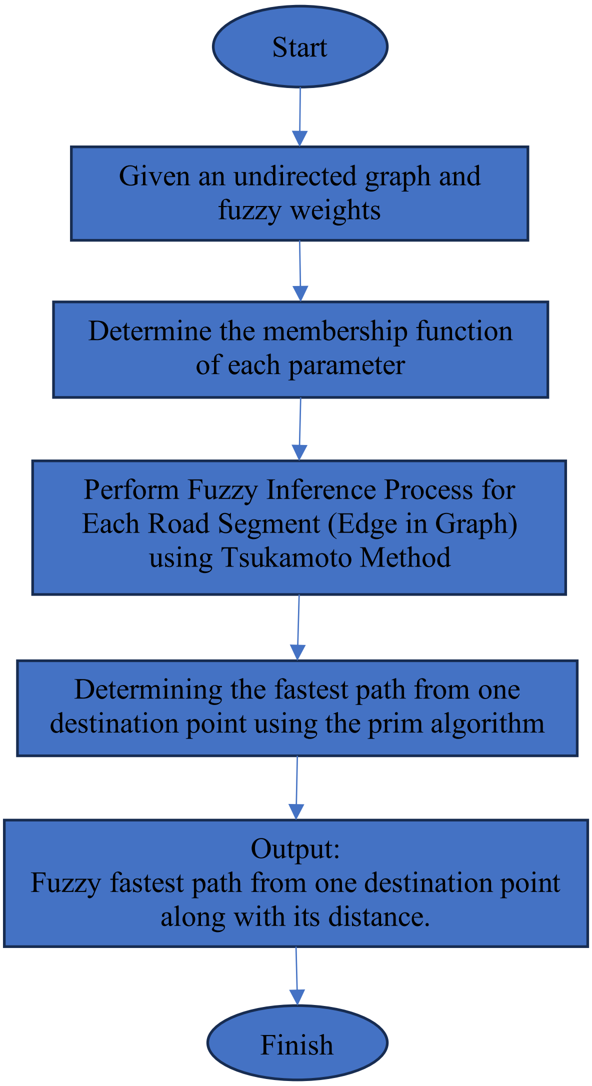


Fig. Research Stages

3. Results and Discussion.

In this section we focus on illustrative examples that emphasize the proposed modified optimal branching algorithm and the modified Prim algorithm. Since FFMST has direct applications in network design, we have considered two numerical examples for directed and undirected fuzzy graphs.

1. Example 1

   To determine the FFMST in the example given below by adopting the modified optimal branching algorithm. Suppose a new neighborhood bank is starting up and will establish its headquarters h, its two branches b1 and b2, and its four ATMs a1, a2, a3, and a4. They have to build a computer network such that the head office, branches, and ATMs can communicate with each other. In addition, they need to build a network with the Federal Bank, f. To know its FFMST is being explained. The feasible network connection costs are listed below using the existing score function and the FFN arithmetic operations are tabulated in Table 3.

   above is given in Figure 1 as a directed FFG $\mathcal{GJ}$. Determining all possible graphs $\mathscr{F}$, an arborescence calculation rooted at the minimum cost *h* is given in Figure 2

Table 1. Feasible Connection Costs

S.No	Edge	Load	Score Function
1	Hf	(0.8,0.4)	0.724
2	Hb_1	(0.4,0.6)	0.424
3	Hb_2	(0.8,0.6)	0.648
4	B_1b_2	(0.7,0.2)	0.6675
5	Fb_1	(0.9,0.3)	0.851
6	Fa_1	(0.6,0.7)	0.4365
7	B_1a_1	(0.5,0.8)	0.3065
8	A_1a_2	(0.8,0.3)	0.7425
9	Ha_2	(0.7,0.5)	0.609
10	B_2a_2	(0.9,0.6)	0.7565
11	B_2a_3	(0.7,0.3)	0.658
12	A_1a_4	(0.5,0.3)	0.549
13	A_3a_4	(0.4,0.5)	0.4695

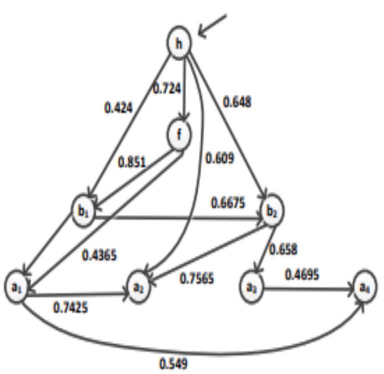


Fig 2.

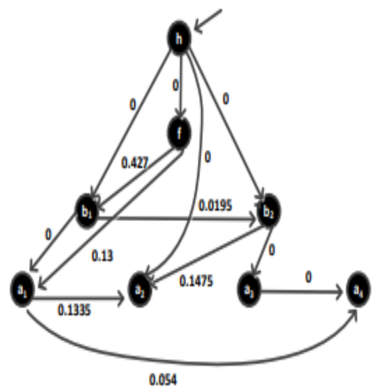


Fig 3.

2. Result 1

   By finding $\mathscr{F}$ FFMST for the directed graph over $\mathcal{GJ}$ given in Figure 3, the minimum cost is found to be 3,839 by modifying the branching algorithm.

https://doi.org/10.31849/digitalzone.v15i2. 22826

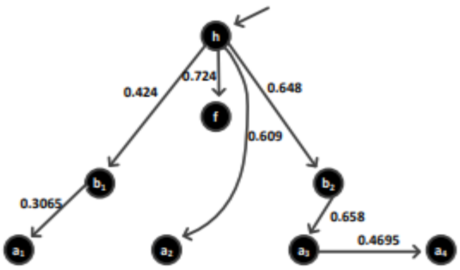


Fig 4. Generating F, FFMST For Directed Graphs

3. Example 2

   The steps to find the fuzzy fastest path in this study are a combination of the fuzzy inference process with the Tsukamoto method and Dijkstra's algorithm. The steps are described as follows:

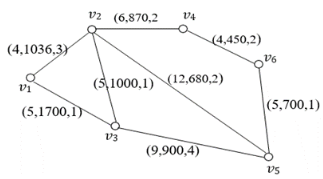
   1. Given a fuzzy weighted undirected graph. The weights on this undirected graph consist of road length, road density, and road conditions.
   2. Determine the membership function for each parameter. The membership function used in this study is a linear membership function.
   3. Performing a fuzzy inference process for each road segment (edge in the graph) using the Tsukamoto method. In the fuzzy inference process, the following rules are used.

Table 2. Evaluation Rules

Road Length	Road Density	Road Condition	Z
Short	Deserted	Good	0.1
Short	Deserted	Broken	0.33
Short	Solid	Good	0.33
Short	Solid	Broken	0.66
Long	Deserted	Good	0.33
Long	Deserted	Broken	0.66
Long	Solid	Good	0.66
Long	Solid	Broken	1

   4. Determine the fastest path from a starting point to a destination point using modifying the optimal branching algorithm. modifying the optimal branching algorithm may not necessarily be used in this step because the calculation result of the fuzzy inference process is a rational number.

Example Given a graph G in Figure 2 that represents a location with a point and a road between two locations with an edge. Each edge is weighted by a fuzzy number, where the first element states the length of the road, the second element states the density of the road, while the third element states the condition of the road. The problem to be solved is which path to take so that the trip can be done as quickly as possible?



Completion Each edge has three parameter values, namely road length, road density, and road condition. In Table 2 below, we will write down the weight of each side.

Table 3. Weight Value of Each Side

Side	Road Length	Road Density	Road Condition
(v_1,v_2)	4	1036	3
(v_1,v_3)	5	1700	1
(v_2,v_3)	5	1000	1
(v_2,v_4)	6	870	2
(v_2,v_5)	12	680	2
(v_3,v_5)	9	900	4
(v_4,v_6)	4	450	2
(v_5,v_6)	5	700	1

Description:
Road condition = 1, means the road condition is damaged
Road condition = 2, means the road condition is good enough
Road condition = 3, means good road condition
Road condition = 4, means very good road condition

In the first step, we determine the membership function for each parameter.

1. Road length membership function

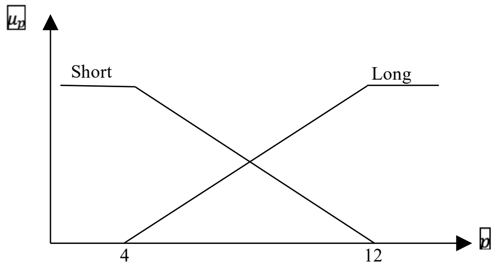


Fig 5: Road Length Membership Function

Fig 5 illustrates the membership function of road length which can be expressed in the following function form:

$$\mu_p \text{Short}[p] = \begin{cases} 1 \,; p \le 4 \\ \dfrac{12-p}{8} \,; 4 \le p \le 12 \\ 0 \,; p \ge 12 \end{cases}$$

$$\mu_p \text{Long}[p] = \begin{cases} 0 \,; p \le 4 \\ \dfrac{p-4}{8} \,; 4 \le p \le 12 \\ 1 \,; p \ge 12 \end{cases}$$

2. Road density membership function

https://doi.org/10.31849/digitalzone.v15i2. *22826*

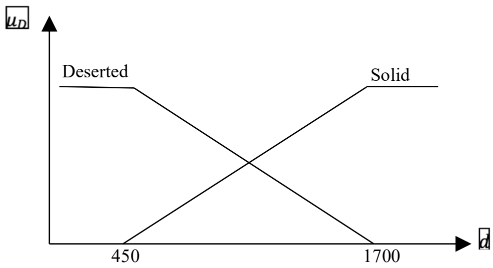


Fig 6: Road Density Membership Function

Fig 6 illustrates the road density membership function which can be expressed in the following function form:

$$\mu_D \text{Deserted}\ [d] = \begin{cases} 1\,;d \le 450 \\ \dfrac{1700-d}{1250}\,;450 \le d \le 1700 \\ 0\,;d \ge 1700 \end{cases}$$

$$\mu_D \text{Solid}\ [d] = \begin{cases} 0\,;d \le 450 \\ \dfrac{d-450}{1250}\,;450 \le d \le 1700 \\ 1\,;d \ge 1700 \end{cases}$$

3. Road condition membership function

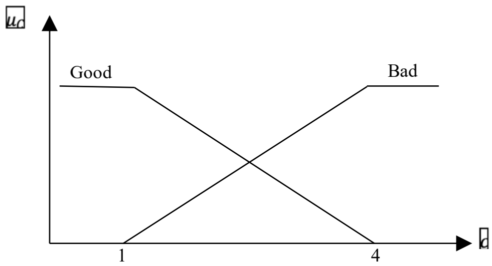


Fig 7: Road Condition Membership Function

Fig 7 illustrates the road condition membership function which can be expressed in the following function form:

$$\mu_C \text{Good}\ [c] = \begin{cases} 1\,;c \le 1 \\ \dfrac{4-c}{3}\,;1 \le c \le 4 \\ 0\,;c \ge 4 \end{cases}$$

$$\mu_C \text{Bad}\ [c] = \begin{cases} 0\,;c \le 1 \\ \dfrac{c-1}{3}\,;1 \le c \le 4 \\ 1\,;c \ge 4 \end{cases}$$

In the second step, perform the fuzzy inference process for each edge. In this discussion, the fuzzy inference process is carried out with the Tsukamoto method.

1. Side (v_1, v_2) with weight (4,1036,3).
   a. Membership value of road length:
      μ_P Short [4] = 1 and μ_P Long [4] = 0
   b. Membership value of road density:
      μ_D Deserted [1036] = $\frac{1700-1036}{1250}$ = 0,5312 and μ_D Solid [1036] = $\frac{1036-450}{1250}$ = 0,4688
   c. Road condition membership value:
      μ_C Good [3] = $\frac{4-3}{3}$ = 0,33 and μ_C Bad [3] = $\frac{3-1}{3}$ = 0,67

Rules 1:

α Predicate 1 = μ_PShort ∩ μ_DDeserted ∩ μ_CGood
$= \min\{\mu_p \text{Short}[4], \mu_D \text{Deserted}[1036], \mu_C \text{Good}[3]\}$
$= \min\{1{,}0.5312{,}0.33\}$
$= 0.33$
Based on Table 5, obtained $z_1 = 0.1$

Rules 2:

α Predicate 2 = μ_PShort ∩ μ_DDeserted ∩ μ_CBad
$= \min\{\mu_p \text{Short}[4], \mu_D \text{Deserted}[1036], \mu_C \text{Bad}[3]\}$
$= \min\{1{,}0.5312{,}0.67\}$
$= 0.5312$
Based on Table 5, obtained $z_2 = 0.33$

Rules 3:

α Predicate 3 = μ_PShort ∩ μ_DSolid ∩ μ_CGood
$= \min\{\mu_p \text{Short}[4], \mu_D \text{Solid}[1036], \mu_C \text{Good}[3]\}$
$= \min\{1{,}0.5312{,}0.33\}$
$= 0.33$
Based on Table 5, obtained $z_3 = 0.33$

Rules 4:

α Predicate 4 = μ_PShort ∩ μ_DSolid ∩ μ_CBad
$= \min\{\mu_p \text{Short}[4], \mu_D \text{Solid}[1036], \mu_C \text{Bad}[3]\}$
$= \min\{1{,}0.5312{,}0.67\}$
$= 0.4688$
Based on Table 5, obtained $z_4 = 0.66$

Rules 5:

α Predicate 5 = μ_PLong ∩ μ_DDeserted ∩ μ_CGood
$= \min\{\mu_p \text{Long}[4], \mu_D \text{Deserted}[1036], \mu_C \text{Good}[3]\}$
$= \min\{0{,}0.5312{,}0.33\}$
$= 0$
Based on Table 5, obtained $z_5 = 0.33$

Rules 6:

α Predicate 6 = μ_PLong ∩ μ_DDeserted ∩ μ_CBad
$= \min\{\mu_p \text{Long}[4], \mu_D \text{Deserted}[1036], \mu_C \text{Bad}[3]\}$
$= \min\{0{,}0.5312{,}0.67\}$
$= 0$
Based on Table 5, obtained $z_6 = 0.66$

Rules 7:

α Predicate 7 = μ_PLong ∩ μ_DSolid ∩ μ_CGood
$= \min\{\mu_p \text{Long}[4], \mu_D \text{Solid}[1036], \mu_C \text{Good}[3]\}$
$= \min\{0{,}0.5312{,}0.33\}$
$= 0$
Based on Table 5, obtained $z_7 = 0.66$

https://doi.org/10.31849/digitalzone.v15i2. 22826

Rules 8:

$$\alpha \text{ Predicate } 8 = \mu_P\text{Long} \cap \mu_D\text{Solid} \cap \mu_C\text{Bad}$$
$$= \min\{\mu_p\text{Long}[4], \mu_D\text{Solid}[1036], \mu_C\text{Bad}[3]\}$$
$$= \min\{0, 0.5312, 0.67\}$$
$$= 0$$

Based on Table 5, obtained $z_8 = 0.66$

Then the final z value is obtained by averaging all the weighted z values.

$$z = \frac{(0.33 \times 0.1) + (0.5312 \times 0.33) + (0.33 \times 0.33) + (0.4688 \times 0.66)}{0.33 + 0.5312 + 0.33 + 0.4688} = 0.37747$$

In the same way, the final z values for the other edges are also obtained as follows:

2. Side (v_1, v_3) with weight (5,1700,1).

$$z = \frac{(0.875 \times 0.33) + (0.125 \times 0.66)}{0.875 + 0.125} = 0{,}37125$$

3. Side (v_2, v_3) with weight (5,1000,1).

$$z = \frac{(0.56 \times 0.1) + (0.44 \times 0.33) + (0.125 \times 0.33) + (0.125 \times 0.66)}{0.56 + 0.44 + 0.125 + 0.125} = 0.25996$$

4. Side (v_2, v_4) with weight (6,870,2).

$$z = 0{,}439036$$

5. Side (v_2, v_5) with weight (12,680,2).

$$z = \frac{(0.66 \times 0.33) + (0.33 \times 0.66) + (0.184 \times 0.66) + (0.184 \times 1)}{0.66 + 0.33 + 0.184 + 0.184} = 0.54568$$

6. Side (v_3, v_5) with weight (9,900,4).

$$z = \frac{(0.375 \times 0.33) + (0.36 \times 0.66) + (0.625 \times 0.66) + (0.36 \times 1)}{0.375 + 0.36 + 0.625 + 0.36} = 0.659215$$

7. Side (v_4, v_6) with weight (4,450,2).

$$z = \frac{(0.66 \times 0.1) + (0.33 \times 0.33)}{0.66 + 0.33} = 0{,}176666$$

8. Side (v_3, v_5) with weight (5,700,1).

$$z = \frac{(0.8 \times 0.1) + (0.2 \times 0.33) + (0.125 \times 0.33) + (0.125 \times 0.66)}{0.8 + 0.2 + 0.125 + 0.125} = 0.2158$$

Step three, draw the graph G again by weighting each remainder with a value of *z*.

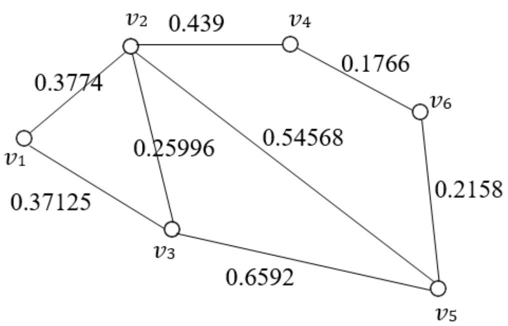


Fig 8: Graph G Weighted by the z Value of Each Edge

The next step is to determine the fastest path from the point v_1 to the point v_6 using the optimal branching algorithm. with iteration results such as Table 2. From Table 2 it can be seen that the fastest trajectory of the $v1$ to $v6$ is $v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_6$.

The proposed approach shows efficiency in managing weight parameters and provides a more accurate representation for ambiguous parameters in graphs, making it more efficient and flexible in solving MST problems. and algorithm can be used to solve the FFMST problem with directed and undirected connected weighted FFG, and the approach can be extended for multi-objective FFMST problems.

The modified algorithms are tested using numerical examples, including a local bank network and a directed graph, and the results show the effectiveness of the proposed approach in solving the FFMST problem.

The results show that the proposed algorithm can be used to find the fastest path in a weighted FFG, and the results of the example problem are presented in Table 2.

Same Rules and Variables If the rules and variables used in Table 2 are the same as those we use, then we can:

a. Compare z values: Directly compare the z values produced by both methods. If there are significant differences, investigate the cause (for example, differences in membership functions, weights, or defuzzification methods).
b. Analyze sensitivity: Change the input values or parameters slightly in both methods and observe the effect on the z value. This can help identify which parts of the model are most sensitive to change.

Different Rules and Variables If the rules and variables used are different, then we need to carry out a more careful analysis. Some things you can do:

a. Identify similarities: Look for similarities between the two methods, for example in terms of basic concepts or goals to be achieved.
b. Compare the results qualitatively: Although the z values may differ numerically, we can compare the general trend or pattern of the results of both methods.
c. Performance evaluation: Compare the performance of both methods in a specific application. For example, if both methods are used to predict traffic, we can compare their prediction accuracy.

4. Conclusion.

In this work we have introduced an algorithmic approach to solve the FFMST problem with directed and undirected connected weighted FFGs. A modified optimal branching algorithm for directed FFGs and a modified Prim algorithm for undirected FFGs are proposed using existing score functions and arithmetic operations of FFNs. We have demonstrated the proposed algorithms using practical examples. In the future the proposed methodology can be executed for large-scale network problems. Furthermore, the proposed algorithm can be extended for multi-objective FFMST which gives scope in shaping the most appropriate results for various problems.

The Tsukamoto approach uses fuzzy rules to determine the edge weights in the minimum spanning tree problem. The edge weights are calculated based on the degree of edge membership in each function. The minimum spanning tree is then found using Prim's algorithm, where the edge with the smallest fuzzy weight is selected first. Both approaches have their own advantages and disadvantages. The Tsukamoto approach is easier to understand and implement, but can produce less than optimal results. The Fermantean approach can produce more optimal results, but is more complex and requires longer computation time.

The most appropriate approach to use depends on the specific needs of the application. If an optimized solution is required and can handle complex problems, Fuzzy FermAnt may be the right choice. However, if a solution that is easy to understand and implement for simpler problems is required, Tsukamoto Fuzzy may be a better choice.

https://doi.org/10.31849/digitalzone.v15i2. 22826

The main findings Modified Prim's Algorithm is effective in solving the Fuzzy Fermatean MST problem for undirected graphs, providing minimum cost results in the case of local bank networks. The modified Optimal Branching algorithm successfully optimizes the network for directed graphs, such as communication between the head office and bank branches.

The proposed algorithm can be used to find the fastest path in a weighted FFG, and the results show that the fastest path from v1 to v6 is v1 → v2 → v4 → v6.

The results show that the modified Prim's algorithm is effective in solving the Fuzzy Fermatean Minimum Spanning Tree (MST) problem for undirected graphs by providing minimum cost results in the case of local bank networks. The proposed algorithm can be extended for multi-objective FFMST which provides scope in forming the most appropriate result for various problems. goal achievement This article presents two entries for solving the operational minimum problem (AGM): one containing Fuzzy Fermatean (FFN) and one containing Fuzzy de Tsukamoto.

The loadedFFN uses a fuzzy Fermatean join to represent pesos in the graph. A modified Prim's algorithm and a modified frequent consequence algorithm are used to deal with the GMS. These modified algorithms use FFN-enabled pons functions and fuzzy arithmetic operations to compare the peso to the graph. Fuzzy de Tsukamoto solving uses a fuzzy inference process to determine the camera more quickly. A fuzzy inference process based on the Tsukamoto method is applied to determine the camera more quickly. Because the two things mentioned above are not worth the responsibility of efficiency and ease of implementation. The missingFFN can produce results more often, but it is more complicated and requires longer computation time. Although the results from Tsukamoto are easier to understand and implement, it can produce smaller results than before. Loading them more than adequately depends on the needs of a particular application.

The Fermatean approach and the Tsukamoto approach both offer solutions to the minimum spanning tree problem, but they differ in terms of complexity and optimality. The Tsukamoto approach, which uses fuzzy rules to determine edge weights based on membership degrees, and then uses Prim's algorithm, is easier to implement but may produce less optimal results. In contrast, the Fermatean approach, although more complex and computationally intensive, can potentially provide a more optimal solution. The best choice depends on the needs of the particular application: prioritize the Tsukamoto approach for simplicity in simpler problems, and the Fermatean approach for optimal solutions in more complex scenarios.

A. Limitations of this study

Although this discussion started out easy to solve, the IFS and PFS still have research limitations such as the following

a. not clear to handle all types of usage
b. presenters who do not start explicitly for the specific forskningen requested in the article

B. Provide Suggestions For Future Research.

Refinancing and Tsukamoto Refinancing are methods used to reduce problems with minimum expenditure (MST) and using omgivelser. Please the method must be useful and useful.Tsukamoto -Making things that are easy and implemented, but can produce optimal results. Fermantean-tilnærmingen can give optimal results, but it is very complex and continuous. Care should be taken in what needs to be done, depending on the specific needs of the application. If optimal learning is the right solution and can handle complex problems, then Fuzzy Fermantean will get the right value. But if this learning is helpful and applied to computer problems, perhaps Tsukamoto Fuzzy has better value.

The following is a comparison between the Fermantean Approach and the Tsukamoto Approach:

Table 4. Comparison between Fermantean and Tsukamoto Approach

Criteria	Fermantean Approach	Tsukamoto Approach

Criteria	Fermantean Approach	Tsukamoto Approach
Basic Principles	The ant algorithm simulates the movement of ants searching for food, leaving a trail of pheromones on the edges it passes.	Fuzzy rules are used to determine the edge weights based on the edge membership level of each function.
Edge weight determination	The pheromone trail on the edge determines the weight of the edge.	The degree of edge membership in each function determines the edge weight.
Search algorithm	The ant algorithm is used to find the minimum spanning tree.	Prim's algorithm is used to find the minimum spanning tree.
Quality of results	Can produce more optimal results.	Less than optimal.
Computation time	More complex and requires longer computation time.	Easier to understand and implement, and faster.
Ease of understanding and implementation	Difficult.	Easy.
Application	Suitable for complex problems with multiple criteria.	Suitable for simpler problems with fewer criteria.

References

[1] Akram, M., & Akmal, A. (2019). Intuitionistic fuzzy graph structures. Journal of Fuzzy Set Theory, 26(3), 233-245. https://doi.org/10.1016/j.fss.2018.09.010
[2] Atanassov, K. (2019). Generalized intuitionistic fuzzy sets and their applications. Fuzzy Sets and Systems, 380(1), 112-121. https://doi.org/10.1016/j.fss.2018.09.012
[3] Bhattacharya, A. (2020). A survey on fuzzy graph theory. Annals of Mathematics and Artificial Intelligence, 88(2), 341-368. https://doi.org/10.1007/s10472-020-09708-w
[4] Senapati, T., & Yager, R. R. (2020). On Fermatean fuzzy sets and their applications in decision making. Journal of Intelligent & Fuzzy Systems, 39(2), 2223-2235. https://doi.org/10.3233/JIFS-2020-1637
[5] Shannon, A., & Atanassov, K. (2020). Intuitionistic fuzzy relations and their applications. Information Sciences, 507, 89-101. https://doi.org/10.1016/j.ins.2019.10.031
[6] Yager, R. R., & Alajlan, N. (2020). On Pythagorean fuzzy sets and their applications in multi-criteria decision making. Fuzzy Sets and Systems, 387, 53-65. https://doi.org/10.1016/j.fss.2019.12.008
[7] Akram, M., Khalid, H., & Yousaf, M. (2021). A study on Fermatean fuzzy graphs with applications in decision-making. Soft Computing, 25(5), 3479-3492. https://doi.org/10.1007/s00500-020-05551-7
[8] Akram, M., & Naz, I. (2021). Intuitionistic fuzzy graphs: An application in shortest path problems. Complexity, 2021, 1-12. https://doi.org/10.1155/2021/6616938
[9] Senapati, T., & Yager, R. R. (2021). On Pythagorean fuzzy graphs with applications to decision making. IEEE Transactions on Fuzzy Systems, 29(4), 793-804. https://doi.org/10.1109/TFUZZ.2020.3013916
[10] Yager, R. R., & Abbasov, A. M. (2022). The theory of Pythagorean fuzzy sets and their applications. Fuzzy Information and Engineering, 14(1), 1-14. https://doi.org/10.1007/s12547-022-00536-6

https://doi.org/10.31849/digitalzone.v15i2. 22826

[11] Senapati, T., & Yager, R. R. (2022). Fermatean fuzzy graph theory and its applications in network design. International Journal of Approximate Reasoning, 145, 47-59. https://doi.org/10.1016/j.ijar.2022.05.003
[12] Yager, R. R., & Petrosyan, A. (2022). Fermatean fuzzy relations: Theory and applications in complex systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(3), 1578-1591. https://doi.org/10.1109/TSMC.2021.3091398
[13] Akram, M., & Shehzad, M. (2023). Applications of Fermatean fuzzy graphs in communication networks. Journal of Applied Research and Technology, 21(2), 89-103. https://doi.org/10.1016/j.jart.2022.12.015
[14] Rosenfeld, A. (2023). Extensions of fuzzy graph theory to Pythagorean fuzzy environments. Journal of Fuzzy Logic and Mathematics, 55(1), 33-45. https://doi.org/10.5391/jflm.2023.55.1.33
[15] Senapati, T., & Yager, R. R. (2024). Decision-making in uncertain environments using Fermatean fuzzy graphs. Decision Support Systems, 156, 113812. https://doi.org/10.1016/j.dss.2023.113812
[16] UPUY, D., & HIARIEY, A. H. (2023). Implementasi Fuzzy Sugeno untuk menentukan Jumlah Produksi Tahu. Jurnal Teknologi Informasi dan Terapan, 10(2), 91-94. https://doi.org/10.25047/jtit.v10i2.327