Design of an Intelligent Mathematics Model for Analyzing Demand, Supply, and Cost Using Fuzzy Logic in MATLAB Environment

Dr. Ayman Abdel-mahmoud Ahmed Imam

Albaha University, Faculty of Science, Mathematics department, KSA

aimam@bu.edu.sa

Abstract: We have in this paper introduced a fuzzy logic intelligent model to investigate the proposed ambiguity relation of demand, supply and cost by using fuzzy logic in MATLAB. To give an efficient support for administrative and economic decision making, by computing fuzzy sub-inputs to look at fuzzy interactions (foggy environments) among these three inputs using uncertainty-based fuzzy inference systems (FIS). The five imaginary sub-inputs to demand are the desire for product, the time, the economy, inventory and production rate, the foggy sub-inputs of costs are production, transportation, storage, energy and seasonal costs, and supply has three imaginary foggy sub-inputs: current inventory, production rate and reliability of supply. These are general fuzzy rules that were used to represent nonlinear input-output relationships, and. The model was put to the test by running scenarios to test how well it predicted and compared performance. This showed that the model was able to deliver true and evolvable information for decision making. It has been useful in understanding the transition from the classical to the contemporary analysis of uncertain data and is therefore very useful for economic decision-makers.

Keywords: Fuzzy Logic - MATLAB - Demand Analysis - Supply Analysis - Cost Analysis - Intelligent Model Design - Fuzzy Inference Systems (FIS) - Economic Decision Support - Data Normalization - Performance Evaluation.

1. Introduction

The more complex the markets and economic landscapes, the more difficult it has been to take decisions about demand, supply and cost. This is because of a lot of factors and because the data for these relationships are not always reliable. (Christopher, 2017) (Taghipour, 2020) (Syntetos, 2009). Analytical models are historically very prescriptive and high in data accuracy, not suitable for dealing with the fluidity and uncertainty of modern markets. Fuzzy logic is a dynamic way of handling ambiguous and uncertain data to represent nonlinear interactions between inputs and outputs. (Bezděk, 2014). This method allows for the combination of different demand, supply and cost determinants (Carlsson, 2012). So here our focus in our paper is create general fuzzy model to simulate this tripartite relation in MATLAB. It relies on writing long fuzzy rules and sub-inputs for all primary unit, like:

- Demand: Depends where product appeal, seasons and economics.
- Supply: Depends on available inventory, volume, supply stability.
- Price: based on production, transport, storage and seasonal costs.

A model of formulated be generic decision support tool and it is evaluated through simulations of fictitious situations that simulate the relationship between demand, supply, and cost under various circumstances.

1.1 Mathematical Formulas and Fuzzy Modelling

Mathematical models are a key way to study economic relationships and management actions. These models had always been based on highly defined assumptions and linear or nonlinear mathematic equations, which restricted their capacity to account for ambiguous or indeterminate data in real economic contexts. Fuzzy analysis is a good technique for dealing with such data among other techniques nowadays. (Korol, 2011) (Forgács, 2021).

In its form, fuzzy logic (spewed out by Zadeh in 1965) explains how uncertainty and impossibility can be mathematically handled by using a non-rigid language. It's used in Fuzzy Inference Systems (FIS) that connect inputs and outputs with "If-Then" statements. (Zadeh, vol. 3 (1971)) (Zadeh L. A., (1996)). This approach has been effective in many instances such as:

• Analyzing economic systems. (Stojić, 2012) (Voynarenko, 2021).

• Risk assessment and decision-making. (Nunes, 2012) (Shapiro, 2015).

• Supply chain management. (Ganga, 2011) (Kumar, 2013).

1.2 Theories of Demand, Supply and Cost

A. Demand

Demand is the desire and capacity of the consumers to buy something or services (Witt, 2001) (Jeffers, 1971). It draws on a number of central economic hypotheses:

• Classical Demand Theory: Measures product cost in terms of demand (demand falls with higher price) (Laibman, 2019).

• Marginal Utility Theory: Proponents of the theory that buying decisions are based on the extra utility gained by the consumer by eating an extra unit of the product (Kauder, 2015).

• Behavioral Theory: Thinks about psychology and society, like reasons and interests (Gavetti, 2012).

B. Supply

Supply is the quantity producers can deliver to the market at given prices (Hall, 1988). The main theories include:

• Standard Supply Theory: Links more supply to higher price (Whinston, 1995).

• Production Efficiency Theory: Focuses on the effect of costs and efficiency on supply (Färe, 1994).

• Current Theories: Interest in supply-impact of technology/supply chain (Lysons, 2020).

C. Cost

Cost is a big factor that has an impact on production and pricing choices (Wu, 2017). Cost-theory Theories:

• Conventional Cost Theory: Separates costs into fixed and variable cost and associates with level of production (Shephard, 2015).

• Economies of Scale Theory: Shows lower costs with more production because fixed costs are spread over more units (Carlino, 2012).

• New Theories: Add in some new costs (e.g., environmental costs, seasonal costs) (De Villiers, 2017).

2. Description of the Submitted Model with Units and Calibration.

2.1 Overview of the Model

The model is a Fuzzy logic model for demand-supply-cost relation. There are three subsystems (demand, supply, cost) of the model that take inputs and turn them into outputs for decision support.

For correctness of analysis, we’ve used a normalization technique to standardize all inputs to a standard range [0-100].

2.2 Demand Analysis System

2.2.1 Sub-Inputs and Measurement Units:

- **Desire for Product:**

Unit: Consumer interest index (0-10).

Calibration: Data are set in the [0-100] range with equation:

$$\text{Normalized Value} = \frac{\text{Input Value}}{10} \times 100$$

- **Time Period:**

Time: Hours or time index (0-24).

Calibration: Times are remapped into [0-100]:

$$\text{Normalized Value} = \frac{\text{Input Value}}{24} \times 100$$

- **Economic Factors:**

Unit: Percentage (0-100).

Calibration: No normalization, all original values used.

- **Purchasing Behavior:**

Unit: Purchase orders count (0-100).

Adjustment: There is no normalization; you can use raw values.

- **Product Availability:**

Unit: Product availability percentage (0-100).

No calibration: Nothing is normalized; value is taken from the raw.

2.2.2 Outputs:

- **Demand Level:** Range: (0-100%).

2.3 Supply Analysis System

2.3.1 Sub-Inputs and Measurement Units:

- **Current Inventory:**

Unit: Number of units (0-10,000 units).

Calibration: Values converted into the [0-100]:

$$\text{Normalized Value} = \frac{\text{Input Value}}{10{,}000} \times 100$$

- **Production Rate:**

Unit: Units/hour (0-500).

Adjustment: Values converted into the range [0-100]:

$$\text{Normalized Value} = \frac{\text{Input Value}}{500} \times 100$$

- **Supply Reliability:**

Unit: Percentage (0-100).

Measurement: No normalization is performed; value is used as-is.

2.3.2 Outputs:

Supply Level: Range: (0-100%).

2.4 Cost Analysis System

2.4.1 Sub-Inputs and Measurement Units:

- **Production Costs:**

1 Unit: Saudi Riyal (SAR) (0-100,000 SAR).

Adjustment: Numbers are updated into [0-100]:

$$\text{Normalized Value} = \frac{\text{Input Value}}{100{,}000} \times 100$$

- **Transportation Costs:**

Unit: SAR (0-20,000 SAR).

Adjustment: Values are rounded up to [0-100]:

$$\text{Normalized Value} = \frac{\text{Input Value}}{20{,}000} \times 100$$

- **Storage Costs:**

Unit: SAR (0-10,000 SAR).

Tests: Reads converted into [0-100]: Values are converted to the value of [0-100]

$$\text{Normalized Value} = \frac{\text{Input Value}}{10{,}000} \times 100$$

- **Seasonal Costs:**

Unit: SAR (0-15,000 SAR).

Adjustment: Values are set to [0-100]:

$$\text{Normalized Value} = \frac{\text{Input Value}}{15{,}000} \times 100$$

- **Energy Costs:**

Unit: SAR (0-10,000 SAR).

Calibration: Numbers are scaled to [0-100]:

$$\text{Normalized Value} = \frac{\text{Input Value}}{10{,}000} \times 100$$

2.4.2 Outputs:

Total Cost: Range: (0-100%).

Integration Between Inputs and Outputs

The sub-inputs for each system are the values that affect the main outputs (demand, supply, cost).

It's the resulting outputs (Demand, Supply, Cost) that estimate the overall performance from a holistic fuzzy model with fuzzy rules.

Calibration As a Critical Module of the Model.

Calibration: Calibration will mix input values from different sources so that they can be compatible with the fuzzy model.

This makes the analysis and the results more precise and effective.

3. Initial Design of the Model (Description Using Functions)

In designing the proposed model, the subsystems (demand, supply, cost) are expressed as mathematical functions that link sub-inputs to main outputs. This approach helps clarify the relationship between inputs and outputs in an organized manner, facilitating the application of fuzzy rules and result analysis.

3.1 Sub models as Mathematical Functions

1. **Demand Model D_{out}:**
    - The demand is expressed as a mathematical function linking sub-inputs:

    $$D_{out} = f(D_1, D_2, D_3, D_4, D_5)$$

    - Where:
        - D_1: Desire for Product.
        - D_2: Time Periods.
        - D_3: Economic Factors.
        - D_4: Purchasing Behavior.
        - D_5: Product Availability.
    - **Output:** D_{out}: Demand Level.
2. **Supply Model S_{out}:**
    - The supply is expressed as a function linking the sub-inputs to the supply level:

    $$S_{out} = f(S_1, S_2, S_3)$$

    - Where:
        - S_1: Current Inventory.
        - S_2: Production Rate.
        - S_3: Supply Reliability.
    - **Output** S_{out}: Supply Level.
3. **Cost Model C_{out}:**
    - The cost is expressed as a mathematical function linking the sub-inputs:

    $$C_{out} = f(C_1, C_2, C_3, C_4, C_5)$$

    - Where:
        - C_1: Production Costs.
        - C_2: Transportation Costs.
        - C_3: Storage Costs.
        - C_4: Seasonal Costs.
        - C_5: Energy Costs.
    - **Output:** C_{out}: Total Cost.

3.2 The Comprehensive Model

- The three subsystems are integrated through their outputs:

$$P = f(D_{\text{out}}, S_{\text{out}}, C_{\text{out}}) \quad \text{Where:}$$

- P: Overall System Performance.
- D_{out} : Demand Level.
- S_{out}: Supply Level.

- C_{out}: Total Cost.

Purpose of Functional Description

- Provide a clear mathematical description that allows for the analysis of relationships between inputs and outputs.
- Facilitate translating the model into fuzzy rules and programming it in the MATLAB environment.

3.2 Designing Fuzzy Rules

3.2.1 Concept of Fuzzy Rules

In the proposed model, fuzzy rules represent the relationship between sub-inputs and outputs using a mathematical methodology based on membership levels (μ\mu) for both inputs and outputs. The rules are divided into three subsystems: demand DD, supply SS, and cost CC, along with a comprehensive performance model PP.

1. **Demand System D_{out}:**
   - The demand level D_{out} is calculated as a mathematical function:

$$D_{out} = f(D_1, D_2, D_3, D_4, D_5)$$

   Where:

   - D_1: Desire for Product.
   - D_2: Time Periods.
   - D_3: Economic Factors.
   - D_4: Purchasing Behavior.
   - D_5: Product Availability.
   - **Fuzzy Rules:**
     - The following rules are used to determine the demand level: If D_1 is High and D_2 is Peak and D_3 is Favorable, then D_{out} is High.

D_1 (Desire)	D_2 (Time)	D_3 (Economic)	D_4 (Purchasing)	D_5 (Availability)	D_{out} (Demand)
Very High	Maximum Peak	Very Favorable	Very Impulsive	Highly Available	Very High
High	Peak	Favorable	Impulsive	Available	High
Medium	Medium	Stable	Moderate	Partially Available	Medium
Low	Sluggish	Weak	Conservative	Limited	Low
Very Low	Minimum Sluggish	Very Weak	Very Conservative	Not Available	Very Low

Table 1: Fuzzy Estimates Table for Demand Sub inputs

2. **Supply System S:**
   - **Mathematical Formulation:**
     - The supply level S_{out} is calculated as: $S_{out} = f(S_1, S_2, S_3)$

   Where:

   - S_1: Current Inventory.

- S_2: Production Rate.
- S_3: Supply Reliability.

- **Fuzzy Rules:**
  - The following rules are used to determine the supply level: If S_1 is High and S_2 is Fast, then S_{out} is High

S_1 (Inventory)	S_2 (Production)	S_3 (Supply)	S_{out} (Supply Level)
Very High	Very Fast	Very Reliable	Very High
High	Fast	Reliable	High
Medium	Medium	Medium	Medium
Low	Slow	Unreliable	Low
Very Low	Very Slow	Very Unreliable	Very Low

Table 2: Fuzzy Estimates Table for Supply Sub inputs

3. **Cost System C:**
   - **Mathematical Formulation:**
     - The total cost C_{out} is calculated as:

$$C_{out} = f(C_1, C_2, C_3, C_4, C_5)$$

Where:

- C_1: Production Costs.
- C_2: Transportation Costs.
- C_3: Storage Costs.
- C_4: Seasonal Costs.
- C_5: Energy Costs.

- **Fuzzy Rules:**
  - The following rules are used to determine the total cost: If C_1 is High and C_2 is High, then C_{out} is High.

C_1 (Production)	C_2 (Transportation)	C_3 (Storage)	C_4 (Seasonal)	C_5 (Energy)	C_{out} (Cost)
Very High	Very High	Very High	Very High	Very High	Very High
High	High	High	High	High	High
Medium	Medium	Medium	Medium	Medium	Medium
Low	Low	Low	Low	Low	Low
Very Low	Very Low	Very Low	Very Low	Very Low	Very Low

Table 3: Fuzzy Estimates Table for Cost Sub inputs

4. **Overall Performance System P:**
   - **Mathematical Formulation:**
     - The performance PP is calculated using the sub-outputs:

$$P = f(D_{out}, S_{out}, C_{out})$$

- **Fuzzy Rules:**
  - The following rules are used to determine performance: If D_{out} is High and S_{out} is Low and C_{out} is High, Then P is Poor.

D_{out} (Demand)	S_{out} (Supply)	C_{out} (Cost)	P_{out} (Performance)
Very High	Very Fast	Very Reliable	Very High
High	Fast	Reliable	High
Balanced (Medium)	Balanced (Medium)	Balanced (Medium)	Balanced (Medium)
Low	Slow	Unreliable	Low
Very Low	Very Slow	Very Unreliable	Very Low

Table 4: Fuzzy Estimates Table for Overall Performance Sub inputs

Note

1. **Complete Mathematical Representation:** Tables are integrated with mathematical formulations illustrating how the rules are designed.
2. **Multiple Levels (5 levels):** Fuzzy levels range from **Very Low** to **Very High**.
3. **Flexibility and Expansion:** Rules can be added or modified based on scenarios.

3.3 Use of Normalization

3.3.1 Importance of Normalization in the Model

- Inputs in the model represent different scales (e.g., number of units, percentages, monetary costs), making it difficult to compare or combine these values directly.
- Normalization aims to unify the range of all sub-inputs to fall within [0-100].
- This ensures consistency of the values used in the fuzzy rules and improves result accuracy.

3.3.2 Normalization Methodology for Each Input

1. **Sub-Inputs for the Demand System DD:**
   - D_1 (Desire for Product)
     - **Unit:** Consumer interest index (0-10).
     - **Normalization:** $D_1^{normalized} = \frac{D_1}{10} \times 100$
   - $D_2(TimePeriods)$
     - **Unit:** Hours (0-24).
     - **Normalization:** $D_2^{\text{normalized}} = \frac{D_2}{24} \times 100$
   - $D_3(EconomicFactors)$
     - **Unit:** Percentage (0-100).
     - **No normalization needed:** $D_3^{\text{normalized}} = D_3$
   - $D_4(PurchasingBehavior)$
     - **Unit:** Percentage (0-100).
     - **No normalization needed:** $D_4^{\text{normalized}} = D_4$
   - $D_5(ProductAvailability)$
     - **Unit:** Percentage (0-100).
     - **No normalization needed:** $D_5^{\text{normalized}} = D_5$
2. **Sub-Inputs for the Supply System \mathbitS\mathbit{S}:**
   - $S_1(CurrentInventory)$
     - **Unit:** Number of units (0-10,000 units).
     - **Normalization:** $S_1^{\text{normalized}} = \frac{S_1}{10{,}000} \times 100$
   - $S_2(ProductionRate)$

- **Unit:** Units/hour (0-500 units).
- **Normalization:** $S_2^{\text{normalized}} = \frac{S_2}{500} \times 100$

- $S_3(SupplyReliability)$
  - **Unit:** Percentage (0-100).
  - **No normalization needed:** $S_3^{\text{normalized}} = S_3$

3. **Sub-Inputs for the Cost System CC:**
   - $C_1(ProductionCosts)$
     - **Unit:** Saudi Riyal (0-100,000 SAR).
     - **Normalization:** $C_1^{\text{normalized}} = \frac{C_1}{100{,}000} \times 100$
   - $C_2(TransportationCosts)$
     - **Unit:** SAR (0-20,000 SAR).
     - **Normalization:** $C_2^{\text{normalized}} = \frac{C_2}{20{,}000} \times 100$
   - $C_3(StorageCosts)$
     - **Unit:** SAR (0-10,000 SAR).
     - **Normalization:** $C_3^{\text{normalized}} = \frac{C_3}{10{,}000} \times 100$
   - $C_4(SeasonalCosts)$
     - **Unit:** SAR (0-15,000 SAR).
     - **Normalization:** $C_4^{\text{normalized}} = \frac{C_4}{15{,}000} \times 100$
   - $C_5(EnergyCosts)$
     - **Unit:** SAR (0-10,000 SAR).
     - **Normalization:** $C_5^{\text{normalized}} = \frac{C_5}{10{,}000} \times 100$

3.3.3 General Normalization Formula

The normalization process for each input xix_i within a specific range $[x_{\min}, x_{\max}]$ can be represented using the equation:

$$x_i^{\text{normalized}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \times 100$$

Importance of Normalization

1. **Consistency of Values:** All values fall within a single range, facilitating their integration into the fuzzy system.
2. **Improved Accuracy:** Helps reduce the impact of significant differences between the scales of different inputs.
3. **Readiness for Programming Application:** Makes inputs ready for implementation in MATLAB without additional processing steps.

4. Implementation of the Model in MATLAB

4.1 Development of the Fuzzy Submodels

The sub models (demand, supply, cost) are implemented using the Fuzzy Logic Toolbox in MATLAB. The steps include:

Practical Steps:

1. **Creating the Fuzzy Model:**

   For each subsystem, a new fuzzy model is created using the newfis command:

```
fis_demand = newfis('DemandModel');  % For the Demand model
fis_supply = newfis('SupplyModel');  % For the Supply model
fis_cost = newfis('CostModel');      % For the Cost model
```

2. **Defining Inputs and Outputs**
   - Inputs are defined using the addvar command with the natural value range for each input.
   - Outputs are defined similarly.

   Example for the Demand Model DD:

```
fis_demand = addvar(fis_demand, 'input', 'DesireForProduct', [0 100]);
fis_demand = addvar(fis_demand, 'input', 'TimePeriod', [0 100]);
fis_demand = addvar(fis_demand, 'output', 'Demand', [0 100]);
```

3. **Defining Membership Functions**
   - For each input and output, five fuzzy levels are defined (Very Low, Low, Medium, High, Very High).
   - Functions like trapmf and trimf are used to create membership functions.

   Example:

```
% Desire for Product
fis_demand = addmf(fis_demand, 'input', 1, 'Very Low', 'trapmf', [0 0 10 20]);
fis_demand = addmf(fis_demand, 'input', 1, 'Low', 'trimf', [15 30 45]);
fis_demand = addmf(fis_demand, 'input', 1, 'Medium', 'trimf', [40 50 60]);
fis_demand = addmf(fis_demand, 'input', 1, 'High', 'trimf', [55 70 85]);
fis_demand = addmf(fis_demand, 'input', 1, 'Very High', 'trapmf', [80 90 100 100]);
```

4. **Adding Fuzzy Rules**
   - The rules are inspired by the previously designed tables.
   - Rules are added using the add rule command.

   Example for the Demand Model:

```
rules = [
    5 5 5 5 5 5 1 1; % If all inputs are Very High, then demand is Very High
    3 3 3 3 3 3 1 1; % If inputs are Medium, then demand is Medium
    1 1 1 1 1 1 1 1; % If inputs are Very Low, then demand is Very Low
];
fis_demand = addrule(fis_demand, rules);
```

Implementing the Supply SS and Cost CC Models

- The same steps are applied to the supply and cost models using their specific inputs and outputs.
- Membership functions and fuzzy inference rules are defined for each model separately.

5. **Integrating Models in MATLAB**

   After creating the submodels, their outputs are assembled into a comprehensive performance model:

```
fis_performance = newfis('PerformanceModel');
fis_performance = addvar(fis_performance, 'input', 'Demand', [0 100]);
fis_performance = addvar(fis_performance, 'input', 'Supply', [0 100]);
fis_performance = addvar(fis_performance, 'input', 'Cost', [0 100]);
fis_performance = addvar(fis_performance, 'output', 'Performance', [0 100]);

% Adding membership functions
fis_performance = addmf(fis_performance, 'input', 1, 'Low', 'trimf', [0 20 40]);
fis_performance = addmf(fis_performance, 'output', 1, 'Excellent', 'trapmf', [60 80 100
100]);

% Adding rules
performance_rules = [
   5 1 1 5 1 1;  % If demand is Very High, supply is Very Low, and cost is Low, then
performance is Excellent
];
fis_performance = addrule(fis_performance, performance_rules);
```

6. **Evaluation and Testing**

   The models are tested using hypothetical values or simulation scenarios.

   Example:

```
input_demand = [85, 90, 70, 80, 95]; % Input values for the Demand model
output_demand = evalfis(input_demand, fis_demand);

input_supply = [75, 80, 85]; % Input values for the Supply model
output_supply = evalfis(input_supply, fis_supply);

input_cost = [50, 60, 55, 45, 40]; % Input values for the Cost model
output_cost = evalfis(input_cost, fis_cost);

% Overall performance
input_performance = [output_demand, output_supply, output_cost];
output_performance = evalfis(input_performance, fis_performance);
```

Results:

- The results are displayed as final fuzzy levels for demand, supply, cost, and overall performance.

4.2 Fuzzy Rules

Rules represent the core of the fuzzy inference process, linking inputs and outputs in a way that considers all possible interactions between different membership levels. In this section, we explain how to formulate the expanded rules for each subsystem and integrate them into the comprehensive model.

1. Expanded Rules for the Demand System D

Rule Formulation

- Fuzzy rules are created to cover all possible combinations of the sub-inputs.
- With 5 levels for each input $(D_1, D_2, D_3, D_4, D_5)$, we can create $5^5 = 3125$ rules.

Example

$$If\ \mu_{D_1} = \text{High},\ \mu_{D_2} = \text{Peak},\ \mu_{D_3} = \text{Favorable},\ \mu_{D_4} = \text{Impulsive},\ \mu_{D_5} = \text{Available},\ \text{Then}\ \mu_{D_{\text{out}}} = \text{Very High}.$$

Implementation in MATLAB

```
rules_demand = [];
for d1 = 1:5
   for d2 = 1:5
      for d3 = 1:5
         for d4 = 1:5
            for d5 = 1:5
               output = max([d1, d2, d3, d4, d5]); % Inference rule
               rules_demand = [rules_demand; d1 d2 d3 d4 d5 output 1 1];
            end
         end
      end
   end
end
fis_demand = addrule(fis_demand, rules_demand);
```

2. Expanded Rules for the Supply System S

Rule Formulation

- For the supply system, with 3 inputs and 5 levels for each input, we can create $5^3 = 125$ rules.

Example

$If\ \mu_{S_1} = \text{High},\ \mu_{S_2} = \text{Fast},\ \mu_{S_3} = \text{Reliable},\ \text{Then}\ \mu_{S_{\text{out}}} = \text{High}.$

Implementation in MATLAB

```
rules_supply = [];
for s1 = 1:5
   for s2 = 1:5
      for s3 = 1:5
         output = min([s1, s2, s3]); % Inference rule
         rules_supply = [rules_supply; s1 s2 s3 output 1 1];
      end
   end
end
fis_supply = addrule(fis_supply, rules_supply);
```

3. Expanded Rules for the Cost System C

Rule Formulation

- With 5 inputs and 5 levels for each input, we can create 55=31255^5 = 3125 rules.

Example

$$If\ \mu_{C_1} = \text{High},\ \mu_{C_2} = \text{High},\ \mu_{C_3} = \text{Medium},\ \mu_{C_4} = \text{Medium},\ \mu_{C_5} = \text{Low},\ \text{Then}\ \mu_{C_{\text{out}}} = \text{High}.$$

Implementation in MATLAB

```
rules_cost = [];
for c1 = 1:5
    for c2 = 1:5
        for c3 = 1:5
            for c4 = 1:5
                for c5 = 1:5
                    output = max([c1, c2, c3, c4, c5]); % Inference rule
                    rules_cost = [rules_cost; c1 c2 c3 c4 c5 output 1 1];
                end
            end
        end
    end
end
fis_cost = addrule(fis_cost, rules_cost);
```

4. Expanded Rules for the Comprehensive Performance System P

Rule Formulation

- With 3 main inputs ($D_{\text{out}}, S_{\text{out}}, C_{\text{out}}$) and 5 levels for each input, we can create $5^3 = 125$ rules.

Example

$$If\ \mu_{D_{\text{out}}} = \text{High},\ \mu_{S_{\text{out}}} = \text{Low},\ \mu_{C_{\text{out}}} = \text{High},\ \text{Then}\ \mu_P = \text{Poor}.$$

Implementation in MATLAB

```
rules_performance = [];
for d_out = 1:5
   for s_out = 1:5
      for c_out = 1:5
         output = max([d_out, s_out, c_out]); % Inference rule
         rules_performance = [rules_performance; d_out s_out c_out output 1 1];
      end
   end
end
fis_performance = addrule(fis_performance, rules_performance);
```

4.3 Challenges and Practical Application

1. **Number of Rules**: As the number of inputs and levels increases, the number of rules grows rapidly, requiring greater computational resources.
2. **Implementation in MATLAB**: MATLAB provides tools to improve performance by accelerating operations using techniques like parfor.

Note:

- Designing expanded rules ensures comprehensive coverage of all possible scenarios, enhancing the model's accuracy.

4.3 Integration Between Sub models and the Comprehensive Model

- The goal of integration is to combine the outputs of the three sub models $(D_{\text{out}}, S_{\text{out}}, C_{\text{out}})$ into a comprehensive performance model PP.
- This integration allows for unified analysis that considers the impact of demand, supply, and cost on the overall performance of the system.

4.3.1 Integration Steps

1. **Unifying Ranges**
   - All sub-outputs $(D_{\text{out}}, S_{\text{out}}, C_{\text{out}})$ are normalized to the range [0-100] to ensure compatibility during integration.
   - This allows for the use of unified fuzzy rules to evaluate overall performance.
2. **Defining Comprehensive Outputs**
   - The overall performance PP is represented as a function dependent on:

$$P = f(D_{\text{out}}, S_{\text{out}}, C_{\text{out}})$$

   Where:

     - D_{out}: Demand Level.
     - S_{out}: Supply Level.
     - C_{out}: Total Cost.
3. **Formulating Fuzzy Rules for Overall Performance**

- The comprehensive fuzzy rules depend on the relationships between demand, supply, and cost.

D_{out} (Demand)	S_{out} (Supply)	C_{out} (Cost)	P_{out} (Performance)
Very High	Very Fast	Very Reliable	Very High
High	Fast	Reliable	High
Balanced (Medium)	Balanced (Medium)	Balanced (Medium)	Balanced (Medium)
Low	Slow	Unreliable	Low
Very Low	Very Slow	Very Unreliable	Very Low

Explanation of the Table

- **First Row**: If demand is **Very High**, supply is **Very Low**, and cost is **Very High**, then performance is **Very Poor**.
- **Third Row**: If all outputs are **Medium**, then performance is **Good**.
- **Fifth Row**: If demand is **Very Low**, supply is **Very High**, and cost is **Very Low**, then performance is **Excellent**.

5. Implementation of Integration in MATLAB

- The performance model is created using MATLAB.
- Adding the three inputs $D_{\text{out}}, S_{\text{out}}, C_{\text{out}}$ and the output PP:

```
fis_performance = newfis('PerformanceModel');
fis_performance = addvar(fis_performance, 'input', 'Demand', [0 100]);
fis_performance = addvar(fis_performance, 'input', 'Supply', [0 100]);
fis_performance = addvar(fis_performance, 'input', 'Cost', [0 100]);
fis_performance = addvar(fis_performance, 'output', 'Performance', [0 100]);
```

5.1 Defining Membership Functions

- Membership functions for the comprehensive output PP are created based on the five levels:

```
fis_performance = addmf(fis_performance, 'output', 1, 'Very Poor', 'trapmf', [0 0 20 40]);
fis_performance = addmf(fis_performance, 'output', 1, 'Poor', 'trimf', [30 50 70]);
fis_performance = addmf(fis_performance, 'output', 1, 'Good', 'trimf', [60 75 90]);
fis_performance = addmf(fis_performance, 'output', 1, 'Excellent', 'trapmf', [80 90 100 100]);
```

5.2 Adding Fuzzy Rules

- Rules are formulated based on the table shown above:

```
rules_performance = [
   5 1 5 1 1; % Rule 1: Demand Very High, Supply Very Low, Cost Very High ->
Performance Very Poor
   4 2 4 2 1; % Rule 2: Demand High, Supply Low, Cost High -> Performance Poor
   3 3 3 3 1; % Rule 3: Demand Medium, Supply Medium, Cost Medium -> Performance
Good
   2 4 2 4 1; % Rule 4: Demand Low, Supply High, Cost Low -> Performance Excellent
   1 5 1 5 1; % Rule 5: Demand Very Low, Supply Very High, Cost Very Low ->
Performance Excellent
];
fis_performance = addrule(fis_performance, rules_performance);
```

5.3 Testing Overall Performance

- The values resulting from the submodels are used to test the overall performance:

```
% Sub-output values
demand_out = 85; % Demand Level
supply_out = 70; % Supply Level
cost_out = 50;   % Cost Level

% Input values to the comprehensive model
performance_out = evalfis([demand_out, supply_out, cost_out], fis_performance);
fprintf('Overall Performance: %.2f\n', performance_out);
```

5.3.1 Integration Results

- The comprehensive model allows performance analysis in a flexible manner that considers the relationships between demand, supply, and cost.
- The results can be used to make strategic decisions to improve system performance.

5.4 Comprehensive Code

This code performs all the required steps; it only asks you to input the sub-input values between 0 and 100.

```
>>>>>>>>>>>>>>>>>>>>>>>>>

% Clear the workspace and command window
clear;
clc;

%% 1. Fuzzy Model for Demand

% Create a new fuzzy inference system for Demand
fis_demand = newfis('DemandModel');
```


10.29322/IJSRP.15.03.2025.p15921

```
% Add input variables for Demand with their membership functions

% 1. Desire For Product
fis_demand = addvar(fis_demand, 'input', 'DesireForProduct', [0 100]);
fis_demand = addmf(fis_demand, 'input', 1, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_demand = addmf(fis_demand, 'input', 1, 'Low', 'trimf', [20 35 50]);
fis_demand = addmf(fis_demand, 'input', 1, 'Medium', 'trimf', [40 55 70]);
fis_demand = addmf(fis_demand, 'input', 1, 'High', 'trimf', [60 75 90]);
fis_demand = addmf(fis_demand, 'input', 1, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 2. Time Period
fis_demand = addvar(fis_demand, 'input', 'TimePeriod', [0 100]);
fis_demand = addmf(fis_demand, 'input', 2, 'VeryShort', 'trapmf', [0 0 10 30]);
fis_demand = addmf(fis_demand, 'input', 2, 'Short', 'trimf', [20 35 50]);
fis_demand = addmf(fis_demand, 'input', 2, 'Medium', 'trimf', [40 55 70]);
fis_demand = addmf(fis_demand, 'input', 2, 'Long', 'trimf', [60 75 90]);
fis_demand = addmf(fis_demand, 'input', 2, 'VeryLong', 'trapmf', [80 90 100 100]);

% 3. Economic Factors
fis_demand = addvar(fis_demand, 'input', 'EconomicFactors', [0 100]);
fis_demand = addmf(fis_demand, 'input', 3, 'VeryUnfavorable', 'trapmf', [0 0 10 30]);
fis_demand = addmf(fis_demand, 'input', 3, 'Unfavorable', 'trimf', [20 35 50]);
fis_demand = addmf(fis_demand, 'input', 3, 'Neutral', 'trimf', [40 55 70]);
fis_demand = addmf(fis_demand, 'input', 3, 'Favorable', 'trimf', [60 75 90]);
fis_demand = addmf(fis_demand, 'input', 3, 'VeryFavorable', 'trapmf', [80 90 100 100]);

% 4. Purchasing Behavior
fis_demand = addvar(fis_demand, 'input', 'PurchasingBehavior', [0 100]);
fis_demand = addmf(fis_demand, 'input', 4, 'VeryConservative', 'trapmf', [0 0 10 30]);
fis_demand = addmf(fis_demand, 'input', 4, 'Conservative', 'trimf', [20 35 50]);
fis_demand = addmf(fis_demand, 'input', 4, 'Average', 'trimf', [40 55 70]);
fis_demand = addmf(fis_demand, 'input', 4, 'Impulsive', 'trimf', [60 75 90]);
fis_demand = addmf(fis_demand, 'input', 4, 'VeryImpulsive', 'trapmf', [80 90 100 100]);

% 5. Activity Periods
fis_demand = addvar(fis_demand, 'input', 'ActivityPeriods', [0 100]);
fis_demand = addmf(fis_demand, 'input', 5, 'VeryInactive', 'trapmf', [0 0 10 30]);
fis_demand = addmf(fis_demand, 'input', 5, 'Inactive', 'trimf', [20 35 50]);
fis_demand = addmf(fis_demand, 'input', 5, 'Moderate', 'trimf', [40 55 70]);
fis_demand = addmf(fis_demand, 'input', 5, 'Active', 'trimf', [60 75 90]);
fis_demand = addmf(fis_demand, 'input', 5, 'VeryActive', 'trapmf', [80 90 100 100]);

% Add output variable for Demand with its membership functions
fis_demand = addvar(fis_demand, 'output', 'Demand', [0 100]);
fis_demand = addmf(fis_demand, 'output', 1, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_demand = addmf(fis_demand, 'output', 1, 'Low', 'trimf', [20 35 50]);
fis_demand = addmf(fis_demand, 'output', 1, 'Medium', 'trimf', [40 55 70]);
fis_demand = addmf(fis_demand, 'output', 1, 'High', 'trimf', [60 75 90]);
```

```
fis_demand = addmf(fis_demand, 'output', 1, 'VeryHigh', 'trapmf', [80 90 100 100]);

% Generate fuzzy rules for Demand
rules_demand = [];
for d1 = 1:5
    for d2 = 1:5
        for d3 = 1:5
            for d4 = 1:5
                for d5 = 1:5
                    output = max([d1, d2, d3, d4, d5]); % Custom logic for rules
                    rules_demand = [rules_demand; d1 d2 d3 d4 d5 output 1 1];
                end
            end
        end
    end
end
fis_demand = addrule(fis_demand, rules_demand);

% Set defuzzification method to 'centroid' (default)
fis_demand.defuzzMethod = 'centroid';

% Request input values for Demand from the user
fprintf('Enter the values for Demand inputs (0-100):\n');
input_desire_for_product = input('  Desire For Product: ');
input_time_period = input('  Time Period: ');
input_economic_factors = input('  Economic Factors: ');
input_purchasing_behavior = input('  Purchasing Behavior: ');
input_activity_periods = input('  Activity Periods: ');

% Fuzzification of inputs for Demand
inputs_demand = [input_desire_for_product, input_time_period, input_economic_factors,
input_purchasing_behavior, input_activity_periods];

% Evaluate Demand value using fuzzy inference system
demand_value = evalfis(inputs_demand, fis_demand);

% Display Demand Value
fprintf('Demand Value: %.2f\n', demand_value);

%% 2. Fuzzy Model for Supply

% Create a new fuzzy inference system for Supply
fis_supply = newfis('SupplyModel');

% Add input variables for Supply with their membership functions

% 1. Current Inventory
fis_supply = addvar(fis_supply, 'input', 'CurrentInventory', [0 100]);
fis_supply = addmf(fis_supply, 'input', 1, 'VeryLow', 'trapmf', [0 0 10 30]);
```

```
fis_supply = addmf(fis_supply, 'input', 1, 'Low', 'trimf', [20 35 50]);
fis_supply = addmf(fis_supply, 'input', 1, 'Medium', 'trimf', [40 55 70]);
fis_supply = addmf(fis_supply, 'input', 1, 'High', 'trimf', [60 75 90]);
fis_supply = addmf(fis_supply, 'input', 1, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 2. Production Rate
fis_supply = addvar(fis_supply, 'input', 'ProductionRate', [0 100]);
fis_supply = addmf(fis_supply, 'input', 2, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_supply = addmf(fis_supply, 'input', 2, 'Low', 'trimf', [20 35 50]);
fis_supply = addmf(fis_supply, 'input', 2, 'Medium', 'trimf', [40 55 70]);
fis_supply = addmf(fis_supply, 'input', 2, 'High', 'trimf', [60 75 90]);
fis_supply = addmf(fis_supply, 'input', 2, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 3. Supply Reliability
fis_supply = addvar(fis_supply, 'input', 'SupplyReliability', [0 100]);
fis_supply = addmf(fis_supply, 'input', 3, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_supply = addmf(fis_supply, 'input', 3, 'Low', 'trimf', [20 35 50]);
fis_supply = addmf(fis_supply, 'input', 3, 'Medium', 'trimf', [40 55 70]);
fis_supply = addmf(fis_supply, 'input', 3, 'High', 'trimf', [60 75 90]);
fis_supply = addmf(fis_supply, 'input', 3, 'VeryHigh', 'trapmf', [80 90 100 100]);

% Add output variable for Supply with its membership functions
fis_supply = addvar(fis_supply, 'output', 'Supply', [0 100]);
fis_supply = addmf(fis_supply, 'output', 1, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_supply = addmf(fis_supply, 'output', 1, 'Low', 'trimf', [20 35 50]);
fis_supply = addmf(fis_supply, 'output', 1, 'Medium', 'trimf', [40 55 70]);
fis_supply = addmf(fis_supply, 'output', 1, 'High', 'trimf', [60 75 90]);
fis_supply = addmf(fis_supply, 'output', 1, 'VeryHigh', 'trapmf', [80 90 100 100]);

% Generate fuzzy rules for Supply
rules_supply = [];
for s1 = 1:5
    for s2 = 1:5
        for s3 = 1:5
            output = max([s1, s2, s3]); % Custom logic for rules
            rules_supply = [rules_supply; s1 s2 s3 output 1 1];
        end
    end
end
fis_supply = addrule(fis_supply, rules_supply);

% Set defuzzification method to 'centroid' (default)
fis_supply.defuzzMethod = 'centroid';

% Request input values for Supply from the user
fprintf('\nEnter the values for Supply inputs (0-100):\n');
input_current_inventory = input('  Current Inventory: ');
input_production_rate = input('  Production Rate: ');
input_supply_reliability = input('  Supply Reliability: ');
```

```
% Fuzzification of inputs for Supply
inputs_supply = [input_current_inventory, input_production_rate, input_supply_reliability];

% Evaluate Supply value using fuzzy inference system
supply_value = evalfis(inputs_supply, fis_supply);

% Display Supply Value
fprintf('Supply Value: %.2f\n', supply_value);

%% 3. Fuzzy Model for Cost

% Create a new fuzzy inference system for Cost
fis_cost = newfis('CostModel');

% Add input variables for Cost with their membership functions

% 1. Production Cost
fis_cost = addvar(fis_cost, 'input', 'ProductionCost', [0 100]);
fis_cost = addmf(fis_cost, 'input', 1, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_cost = addmf(fis_cost, 'input', 1, 'Low', 'trimf', [20 35 50]);
fis_cost = addmf(fis_cost, 'input', 1, 'Medium', 'trimf', [40 55 70]);
fis_cost = addmf(fis_cost, 'input', 1, 'High', 'trimf', [60 75 90]);
fis_cost = addmf(fis_cost, 'input', 1, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 2. Transportation Cost
fis_cost = addvar(fis_cost, 'input', 'TransportationCost', [0 100]);
fis_cost = addmf(fis_cost, 'input', 2, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_cost = addmf(fis_cost, 'input', 2, 'Low', 'trimf', [20 35 50]);
fis_cost = addmf(fis_cost, 'input', 2, 'Medium', 'trimf', [40 55 70]);
fis_cost = addmf(fis_cost, 'input', 2, 'High', 'trimf', [60 75 90]);
fis_cost = addmf(fis_cost, 'input', 2, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 3. Storage Cost
fis_cost = addvar(fis_cost, 'input', 'StorageCost', [0 100]);
fis_cost = addmf(fis_cost, 'input', 3, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_cost = addmf(fis_cost, 'input', 3, 'Low', 'trimf', [20 35 50]);
fis_cost = addmf(fis_cost, 'input', 3, 'Medium', 'trimf', [40 55 70]);
fis_cost = addmf(fis_cost, 'input', 3, 'High', 'trimf', [60 75 90]);
fis_cost = addmf(fis_cost, 'input', 3, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 4. Seasonal Cost
fis_cost = addvar(fis_cost, 'input', 'SeasonalCost', [0 100]);
fis_cost = addmf(fis_cost, 'input', 4, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_cost = addmf(fis_cost, 'input', 4, 'Low', 'trimf', [20 35 50]);
fis_cost = addmf(fis_cost, 'input', 4, 'Medium', 'trimf', [40 55 70]);
fis_cost = addmf(fis_cost, 'input', 4, 'High', 'trimf', [60 75 90]);
fis_cost = addmf(fis_cost, 'input', 4, 'VeryHigh', 'trapmf', [80 90 100 100]);
```

```
% 5. Energy Cost
fis_cost = addvar(fis_cost, 'input', 'EnergyCost', [0 100]);
fis_cost = addmf(fis_cost, 'input', 5, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_cost = addmf(fis_cost, 'input', 5, 'Low', 'trimf', [20 35 50]);
fis_cost = addmf(fis_cost, 'input', 5, 'Medium', 'trimf', [40 55 70]);
fis_cost = addmf(fis_cost, 'input', 5, 'High', 'trimf', [60 75 90]);
fis_cost = addmf(fis_cost, 'input', 5, 'VeryHigh', 'trapmf', [80 90 100 100]);

% Add output variable for Cost with its membership functions
fis_cost = addvar(fis_cost, 'output', 'Cost', [0 100]);
fis_cost = addmf(fis_cost, 'output', 1, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_cost = addmf(fis_cost, 'output', 1, 'Low', 'trimf', [20 35 50]);
fis_cost = addmf(fis_cost, 'output', 1, 'Medium', 'trimf', [40 55 70]);
fis_cost = addmf(fis_cost, 'output', 1, 'High', 'trimf', [60 75 90]);
fis_cost = addmf(fis_cost, 'output', 1, 'VeryHigh', 'trapmf', [80 90 100 100]);

% Generate fuzzy rules for Cost
rules_cost = [];
for c1 = 1:5
    for c2 = 1:5
        for c3 = 1:5
            for c4 = 1:5
                for c5 = 1:5
                    output = max([c1, c2, c3, c4, c5]); % Custom logic for rules
                    rules_cost = [rules_cost; c1 c2 c3 c4 c5 output 1 1];
                end
            end
        end
    end
end
fis_cost = addrule(fis_cost, rules_cost);

% Set defuzzification method to 'centroid' (default)
fis_cost.defuzzMethod = 'centroid';

% Request input values for Cost from the user
fprintf('\nEnter the values for Cost inputs (0-100):\n');
input_production_cost = input('  Production Cost: ');
input_transportation_cost = input('  Transportation Cost: ');
input_storage_cost = input('  Storage Cost: ');
input_seasonal_cost = input('  Seasonal Cost: ');
input_energy_cost = input('  Energy Cost: ');

% Fuzzification of inputs for Cost
inputs_cost = [input_production_cost, input_transportation_cost, input_storage_cost,
input_seasonal_cost, input_energy_cost];

% Evaluate Cost value using fuzzy inference system
cost_value = evalfis(inputs_cost, fis_cost);
```

```
% Display Cost Value
fprintf('Cost Value: %.2f\n', cost_value);

%% 4. Fuzzy Model for Performance

% Create a new fuzzy inference system for Performance
fis_performance = newfis('PerformanceModel');

% Add input variables for Performance with their membership functions

% 1. Demand
fis_performance = addvar(fis_performance, 'input', 'Demand', [0 100]);
fis_performance = addmf(fis_performance, 'input', 1, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_performance = addmf(fis_performance, 'input', 1, 'Low', 'trimf', [20 35 50]);
fis_performance = addmf(fis_performance, 'input', 1, 'Medium', 'trimf', [40 55 70]);
fis_performance = addmf(fis_performance, 'input', 1, 'High', 'trimf', [60 75 90]);
fis_performance = addmf(fis_performance, 'input', 1, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 2. Supply
fis_performance = addvar(fis_performance, 'input', 'Supply', [0 100]);
fis_performance = addmf(fis_performance, 'input', 2, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_performance = addmf(fis_performance, 'input', 2, 'Low', 'trimf', [20 35 50]);
fis_performance = addmf(fis_performance, 'input', 2, 'Medium', 'trimf', [40 55 70]);
fis_performance = addmf(fis_performance, 'input', 2, 'High', 'trimf', [60 75 90]);
fis_performance = addmf(fis_performance, 'input', 2, 'VeryHigh', 'trapmf', [80 90 100 100]);

% 3. Cost
fis_performance = addvar(fis_performance, 'input', 'Cost', [0 100]);
fis_performance = addmf(fis_performance, 'input', 3, 'VeryLow', 'trapmf', [0 0 10 30]);
fis_performance = addmf(fis_performance, 'input', 3, 'Low', 'trimf', [20 35 50]);
fis_performance = addmf(fis_performance, 'input', 3, 'Medium', 'trimf', [40 55 70]);
fis_performance = addmf(fis_performance, 'input', 3, 'High', 'trimf', [60 75 90]);
fis_performance = addmf(fis_performance, 'input', 3, 'VeryHigh', 'trapmf', [80 90 100 100]);

% Add output variable for Performance with its membership functions
fis_performance = addvar(fis_performance, 'output', 'Performance', [0 100]);
fis_performance = addmf(fis_performance, 'output', 1, 'VeryPoor', 'trapmf', [0 0 10 30]);
fis_performance = addmf(fis_performance, 'output', 1, 'Poor', 'trimf', [20 35 50]);
fis_performance = addmf(fis_performance, 'output', 1, 'Average', 'trimf', [40 55 70]);
fis_performance = addmf(fis_performance, 'output', 1, 'Good', 'trimf', [60 75 90]);
fis_performance = addmf(fis_performance, 'output', 1, 'Excellent', 'trapmf', [80 90 100 100]);

% Generate fuzzy rules for Performance
rules_performance = [];
for d = 1:5
    for s = 1:5
        for c = 1:5
            % Custom logic for rules
```

```
            if d >= 4 && s >= 4 && c <= 2
                output = 5; % Excellent
            elseif d >= 3 && s >= 3 && c <= 3
                output = 4; % Good
            elseif d == s
                output = 3; % Average
            else
                output = min([d, s, 6 - c]); % Adjusting based on cost
            end
            rules_performance = [rules_performance; d s c output 1 1];
        end
    end
end
fis_performance = addrule(fis_performance, rules_performance);

% Set defuzzification method to 'centroid' (default)
fis_performance.defuzzMethod = 'centroid';

% Evaluate Performance value using fuzzy inference system
inputs_performance = [demand_value, supply_value, cost_value];
performance_value = evalfis(inputs_performance, fis_performance);

% Display Performance Value
fprintf('\nPerformance Value: %.2f\n', performance_value);

%% 5. Plotting the Input and Output Values

% Plot 1: Variation among Demand sub-inputs
figure;
bars_demand = [input_desire_for_product, input_time_period, input_economic_factors,
input_purchasing_behavior, input_activity_periods];
bar(bars_demand);
title('Variation among Demand Sub-Inputs');
xlabel('Demand Sub-Inputs');
ylabel('Value');
set(gca, 'XTickLabel', {'Desire', 'Time', 'Economic', 'Purchasing', 'Activity'});
set(gca, 'XTickLabelRotation', 45);

% Plot 2: Variation among Supply sub-inputs
figure;
bars_supply = [input_current_inventory, input_production_rate, input_supply_reliability];
bar(bars_supply);
title('Variation among Supply Sub-Inputs');
xlabel('Supply Sub-Inputs');
ylabel('Value');
set(gca, 'XTickLabel', {'Inventory', 'Production', 'Reliability'});
set(gca, 'XTickLabelRotation', 45);

% Plot 3: Variation among Cost sub-inputs
```

```
figure;
bars_cost = [input_production_cost, input_transportation_cost, input_storage_cost, input_seasonal_cost, input_energy_cost];
bar(bars_cost);
title('Variation among Cost Sub-Inputs');
xlabel('Cost Sub-Inputs');
ylabel('Value');
set(gca, 'XTickLabel', {'Production', 'Transport', 'Storage', 'Seasonal', 'Energy'});
set(gca, 'XTickLabelRotation', 45);

% Plot 4: Variation among Demand, Supply, Cost, and Performance
figure;
bars_overall = [demand_value, supply_value, cost_value, performance_value];
bar(bars_overall);
title('Variation among Demand, Supply, Cost, and Performance');
xlabel('Main Outputs');
ylabel('Value');
set(gca, 'XTickLabel', {'Demand', 'Supply', 'Cost', 'Performance'});
set(gca, 'XTickLabelRotation', 45);

%% End of Code
```

Note: Ensure you have the Fuzzy Logic Toolbox installed in MATLAB to run this code successfully.

6: Implementing the Fuzzification Process in the Proposed Model

6.1 Introduction

In this chapter, we provide a detailed explanation of how the fuzzification process is applied in the proposed model for analyzing demand, supply, and cost. We focus on how numerical input values are transformed into membership degrees of predefined fuzzy sets and how these degrees are used in the fuzzy inference system to arrive at the final results.

6.2 Fuzzification in the Model

6.2.1 Identifying Input and Output Variables

The model is designed to handle a set of input variables for each of the three sub-models:

- **Demand Model:**
    - Desire for the product
    - Time periods
    - Economic factors
    - Purchasing behavior
    - Activity periods
- **Supply Model:**
    - Current inventory
    - Production rate

    - Supply reliability
- **Cost Model:**
    - Production costs
    - Transportation costs
    - Storage costs
    - Seasonal costs
    - Energy costs

For each of these variables, the value range is defined between 0 and 100.

6.2.2 Defining Membership Functions for Inputs

For each input variable, a set of fuzzy membership functions representing linguistic terms (such as "Very Low," "Low," "Medium," "High," "Very High") is defined. These functions are used to convert numerical input values into membership degrees.

Example for the variable "Desire for the Product" in the Demand Model:

- **Very Low:** Trapezoidal function (trapmf) with breakpoints at [0, 0, 20, 40].
- **Low:** Triangular function (trimf) with peak points at [20, 40, 60].
- **Medium:** Triangular function with peak points at [40, 60, 80].
- **High:** Triangular function with peak points at [60, 80, 100].
- **Very High:** Trapezoidal function with breakpoints at [80, 100, 100, 100].

6.2.3 Converting Input Values to Membership Degrees

When a numerical value is entered for any variable, its membership degree for each fuzzy set is calculated using the defined membership functions.

Practical Example:

Assume the input value for "Desire for the Product" is **70**.

Calculating Membership Degrees:

1. **Very Low:**
    - Since 70 is greater than 40, the membership degree is 0.
2. **Low:**
    - Since 70 is greater than 60, the membership degree is 0.
3. **Medium:**
    - The triangular function for "Medium" is between 40 and 80.
    - Since 70 falls between 60 and 80, we calculate the membership degree:

$$\mu_{\text{Medium}}(70) = \frac{80 - 70}{20} = 0.5$$

4. **High:**
    - The triangular function for "High" is between 60 and 100.
    - Since 70 falls between 60 and 80, we calculate the membership degree:

$$\mu_{\text{High}}(70) = \frac{70 - 60}{20} = 0.5$$

5. **Very High:**
   - Since 70 is less than 80, the membership degree is 0.

Result:

- **Medium:** Membership degree = 0.5
- **High:** Membership degree = 0.5
- Other sets: Membership degree = 0

This means that the value 70 partially belongs to both the "Medium" and "High" sets by 50% each.

6.2.4 Applying the Process to All Inputs

The same steps are applied to all input variables in the three sub-models, resulting in a set of membership degrees for each variable.

6.3 Applying the Fuzzy Inference System

After obtaining the membership degrees for the inputs, we proceed to apply the fuzzy inference rules to determine the outputs.

6.3.1 Formulating Fuzzy Rules

Fuzzy rules link inputs and outputs using "IF-THEN" statements.

Example of a rule in the Demand Model:

- **IF the desire for the product is "High" AND the time periods are "Medium," THEN the demand is "High."**

6.3.2 Evaluating the Rules

For each rule, we perform the following:

1. **Identify the membership degrees of the inputs in the rule.**
2. **Calculate the rule's activation degree:**
   - We use the fuzzy "AND" operation, typically the minimum (MIN) function.
   - The activation degree is the minimum of the input membership degrees in the rule.
3. **Apply the activation degree to the associated output.**

Example:

Using the previous membership degrees for "Desire for the Product" and an assumed value of 65 for "Time Periods":

- **Time Periods:**
    - Using membership functions, assume the membership degree for "Medium" is 0.25 and for "High" is 0.75.

Rule Evaluation:

- Membership degree for "Desire for the Product" = 0.5 ("High")
- Membership degree for "Time Periods" = 0.75 ("High")
- Activation degree of the rule = MIN(0.5, 0.75) = 0.5
- Apply the activation degree (0.5) to the output "Demand" in the "High" set.

6.3.3 Aggregating Fuzzy Outputs

After evaluating all the rules, we may obtain multiple membership degrees for the same output in different sets. We aggregate these outputs using the fuzzy "OR" operation, typically the maximum (MAX) function.

Example:

- "Demand" in the "Medium" set: Membership degrees from different rules might be [0.3, 0.4].
    - After aggregation: Membership degree = MAX(0.3, 0.4) = 0.4
- "Demand" in the "High" set: Membership degrees [0.5, 0.6].
    - After aggregation: Membership degree = MAX(0.5, 0.6) = 0.6

6.4 Defuzzification to Obtain Final Results

To convert the fuzzy outputs into numerical values that can be used in decision-making, we use the defuzzification process.

6.4.1 Centroid Method

The centroid method is one of the most common defuzzification techniques. It relies on calculating the weighted average of the outputs.

Formula:

$$\text{Output Value} = \frac{\int_{\text{domain}} \mu_{\text{output}}(x) \times x \, dx}{\int_{\text{domain}} \mu_{\text{output}}(x) \, dx}$$

Where:

- $\mu_{\text{output}}(x)$ is the membership function of the output.
- x is the independent variable over the output domain.

6.4.2 Calculating the Final Output Value

ISSN 2250-3153

Using the aggregated membership degrees and the membership functions of the different sets, we calculate the final output value.

Example:

- Membership degrees for the output "Demand":
  - "Low": 0.0
  - "Medium": 0.4
  - "High": 0.6
  - "Very High": 0.2
- We use the membership functions for the sets "Medium," "High," and "Very High" to calculate the centroid.

Result:

- The final value of demand might be, for example, 72.5.

6.5 Applying the Process to Other Sub-Models

The same steps are applied to the Supply and Cost models:

- **Calculate membership degrees for the inputs.**
- **Evaluate the fuzzy rules specific to each model.**
- **Aggregate the fuzzy outputs.**
- **Defuzzify to obtain the final output values.**

6.6 Integration into the Overall Performance Model

After obtaining the final values for demand, supply, and cost, we use them as inputs to the overall performance model.

6.6.1 Fuzzifying the Input Values for Performance

- We convert the numerical values of demand, supply, and cost into membership degrees of the predefined fuzzy sets in the performance model.

6.6.2 Applying Performance Rules

- We apply fuzzy inference rules that link demand, supply, and cost to determine the performance level.

Example of a rule in the Performance Model:

- **IF demand is "High" AND supply is "Low" AND cost is "High," THEN performance is "Poor."**

6.6.3 Defuzzifying to Obtain the Performance Value

- After aggregating the fuzzy outputs for performance, we use the defuzzification method to get the final performance value.

6.7 Importance of the Fuzzification Process in the Model

- **Representing Uncertainty:** The fuzzification process helps handle uncertain or ambiguous data.
- **Model Flexibility:** It allows the model to be applied to a wide range of values and conditions.
- **Decision Support:** It provides more accurate and reliable results to support strategic decisions.

6.8 Conclusion

The fuzzification process represents a critical step in the proposed model, as it allows the transformation of numerical input values into fuzzy information that can be processed using a fuzzy inference system. By applying this process, we were able to handle the complexities and uncertainties in real-world data, leading to final results that effectively support the decision-making process.

In summary, this approach provides a robust framework for analyzing complex relationships among various variables and enhances the ability of organizations to respond to dynamic changes in the business environment.

7. Simulation and In-Depth Analysis

7.1 Introduction

In this chapter, we conduct a comprehensive simulation of the intelligent model developed for analyzing demand, supply, and cost using fuzzy logic in MATLAB. The simulation aims to evaluate the effectiveness of the model in predicting and analyzing performance under various scenarios that represent realistic market conditions. By applying different inputs, we can observe how the model responds to changes in demand, supply, and cost factors, providing valuable insights for strategic decision-making.

7.2 Simulation Objectives

- **Evaluate Model Accuracy:** Verify the model's ability to produce accurate results under different conditions.
- **Analyze Input Sensitivity:** Understand the impact of changes in sub-inputs on the main outputs.
- **Test Model Flexibility:** Ensure the model can handle uncertainty and fluctuations in data effectively.

7.3 Scenarios and Results

We have designed three hypothetical scenarios covering a wide range of market conditions:

Scenario 1: High Demand, Medium Supply, High Cost

Inputs:

- **Demand Inputs:**
  - Desire for Product: 90
  - Time Periods: 80
  - Economic Factors: 85
  - Purchasing Behavior: 75
  - Product Availability: 60
- **Supply Inputs:**
  - Current Inventory: 30
  - Production Rate: 40
  - Supply Reliability: 50
- **Cost Inputs:**
  - Production Costs: 80
  - Transportation Costs: 70
  - Storage Costs: 75
  - Seasonal Costs: 85
  - Energy Costs: 80

Results:

- **Demand Level (D):** 82.63
- **Supply Level (S):** 50.00
- **Total Cost (C):** 80.89
- **Overall Performance (P):** 80.31

Scenario 2: Medium Demand, Medium Supply, Medium Cost

Inputs:

- **Demand Inputs:**
  - Desire for Product: 50
  - Time Periods: 51
  - Economic Factors: 52
  - Purchasing Behavior: 49
  - Product Availability: 50
- **Supply Inputs:**
  - Current Inventory: 54
  - Production Rate: 48
  - Supply Reliability: 52
- **Cost Inputs:**
  - Production Costs: 51
  - Transportation Costs: 55
  - Storage Costs: 49
  - Seasonal Costs: 48
  - Energy Costs: 55

Results:

- **Demand Level (D):** 50.96
- **Supply Level (S):** 52.89
- **Total Cost (C):** 53.44
- **Overall Performance (P):** 52.20

Scenario 3: High Demand, High Supply, Low Cost

Inputs:

- **Demand Inputs:**
  - Desire for Product: 30
  - Time Periods: 20
  - Economic Factors: 25
  - Purchasing Behavior: 35
  - Product Availability: 80
- **Supply Inputs:**
  - Current Inventory: 80
  - Production Rate: 85
  - Supply Reliability: 90
- **Cost Inputs:**
  - Production Costs: 20
  - Transportation Costs: 15
  - Storage Costs: 25
  - Seasonal Costs: 30
  - Energy Costs: 20

Results:

- **Demand Level (D):** 80.00
- **Supply Level (S):** 82.63
- **Total Cost (C):** 27.32
- **Overall Performance (P):** 80.27

7.4 In-Depth Interpretation and Analysis of Results

Scenario 1: High Demand, Medium Supply, High Cost

Detailed Analysis:

1. **Demand-Supply Gap:**
   - **High Demand (82.63) vs. Medium Supply (50.00):**
     - The model indicates a significant demand that surpasses the supply capacity.
     - This gap could result in unmet customer needs, leading to potential loss of sales and customer dissatisfaction.
2. **Supply Constraints:**
   - **Low Inventory and Production Rate:**
     - Current inventory (30) and production rate (40) are insufficient to meet high demand.

        - Supply reliability is average (50), which may not mitigate supply issues.

3. **High Costs (80.89):**
    - Elevated production, transportation, storage, seasonal, and energy costs.
    - High costs can erode profit margins and limit competitive pricing strategies.
4. **Overall Performance (80.31):**
    - Despite supply and cost challenges, the performance score is high.
    - This may be attributed to the strong demand driving overall business performance.

Strategic Implications:

- **Enhance Production Capacity:**
    - Invest in scaling up production facilities or outsourcing to meet demand.
    - Implement process improvements to increase efficiency.
- **Optimize Supply Chain:**
    - Improve inventory management through demand forecasting.
    - Strengthen relationships with suppliers to enhance reliability.
- **Cost Reduction Initiatives:**
    - Review operations to identify areas for cost savings.
    - Negotiate better terms with suppliers and logistics providers.
- **Revenue Maximization:**
    - Consider dynamic pricing strategies to maximize revenue.
    - Offer premium products or services to capitalize on high demand.

Scenario 2: Medium Demand, Medium Supply, Medium Cost

Detailed Analysis:

1. **Market Equilibrium:**
    - **Demand (50.96) and Supply (52.89) Levels:**
        - The close alignment suggests a balanced market with no significant shortages or surpluses.
        - This stability allows for predictable operations.
2. **Moderate Costs (53.44):**
    - Costs are manageable, enabling sustainable operations.
    - Provides an opportunity to focus on efficiency improvements.
3. **Overall Performance (52.20):**
    - Indicates steady but unspectacular performance.
    - Reflects a mature market with limited growth opportunities.

Strategic Implications:

- **Market Differentiation:**
    - Innovate products or services to stand out in the market.
    - Enhance brand value through marketing and customer engagement.
- **Operational Excellence:**
    - Implement continuous improvement methodologies like Lean or Six Sigma.
    - Invest in employee training to boost productivity.
- **Explore New Markets:**

- Consider expanding into emerging markets or niches.
- Diversify product offerings to stimulate demand.

Scenario 3: High Demand, High Supply, Low Cost

Detailed Analysis:

1. **High Demand (80.00) Despite Low Desire Indicators:**
    - The model output shows high demand, which may seem contradictory given low desire for product (30) and unfavorable economic factors (25).
    - High product availability (80) could be significantly influencing demand.
2. **Robust Supply Capacity (82.63):**
    - High inventory levels and production rates indicate readiness to meet demand.
    - Supply reliability is very high (90), ensuring consistent product availability.
3. **Low Costs (27.32):**
    - Efficient operations lead to reduced costs across the board.
    - This cost advantage can be leveraged for competitive pricing.
4. **Overall Performance (80.27):**
    - High performance suggests successful alignment of supply capabilities with demand, coupled with cost efficiency.

Strategic Implications:

- **Investigate Demand Drivers:**
    - Analyze factors contributing to high demand despite low initial indicators.
    - Possible factors could include market trends, competitor issues, or latent customer needs.
- **Leverage Cost Leadership:**
    - Utilize low costs to offer attractive pricing or increase marketing efforts.
    - Invest in customer acquisition strategies to boost demand further.
- **Prevent Overstocking Risks:**
    - Monitor inventory levels to avoid excess stock and associated costs.
    - Adjust production schedules based on accurate demand forecasting.

7.5 Comparative Insights and Recommendations

Key Observations:

- **Data Integrity Importance:**
    - Scenario 3 highlights the need to ensure data accuracy and model validation.
    - Discrepancies between inputs and outputs necessitate a review of the model or data collection methods.
- **Cost Impact on Performance:**
    - High costs in Scenario 1 limit profitability, while low costs in Scenario 3 enhance competitive positioning.
    - Effective cost management is crucial for sustaining and improving performance.
- **Demand and Supply Alignment:**
    - Scenarios demonstrate the significance of aligning supply capabilities with market demand.

- Mismatches can lead to lost opportunities or wasted resources.

Overall Strategic Recommendations:

1. **Continuous Monitoring and Analysis:**
    - Regularly use the model to assess market conditions and operational performance.
    - Adjust strategies promptly in response to changing market dynamics.
2. **Enhance Data Accuracy:**
    - Implement robust data collection and validation processes.
    - Use real-time data analytics for more accurate forecasting and decision-making.
3. **Optimize Operations:**
    - Invest in technologies and practices that improve efficiency and reduce costs.
    - Foster a culture of continuous improvement within the organization.
4. **Market Adaptability:**
    - Develop flexible business models that can adapt to market fluctuations.
    - Diversify products and markets to mitigate risks.
5. **Customer-Centric Strategies:**
    - Focus on understanding customer needs and preferences.
    - Enhance customer engagement and satisfaction to build brand loyalty.

The simulation results demonstrate the intelligent model's capability to analyze complex interdependencies among demand, supply, and cost factors. By applying different scenarios, we observe how varying inputs affect overall performance, providing valuable insights for strategic planning. The in-depth analysis emphasizes the importance of aligning operational capabilities with market demands, managing costs effectively, and maintaining data integrity. Implementing the recommendations derived from this analysis can enhance decision-making processes and drive sustainable business growth.

8. Recommendations and General Conclusions

8.1 Recommendations

1. **Define Inputs Based on Real Factors**
    - **Customize Sub-Inputs:** Since the sub-inputs used in this study are hypothetical and were created solely for demonstration purposes, it is essential for organizations to define sub-inputs according to the real factors that influence demand, supply, and cost in their specific context. This is one of the most critical recommendations to ensure the model's accuracy and effectiveness when applied practically.
    - **Analyze Influential Factors:** Conduct comprehensive studies to identify the most impactful factors on the organization's operations, ensuring the model accurately reflects practical reality.
2. **Enhance Data Accuracy and Improve Data Collection**
    - **Implement Modern Data Collection Techniques:** Utilize integrated information systems and sensing technologies to obtain more precise, real-time data.
    - **Train on Data Quality:** Educate teams on the importance of data accuracy and how to verify data integrity before inputting it into the model.

3. **Develop the Model and Adapt It to Real Data**
   - **Test the Model with Real Data:** Apply the model to actual market data to improve its accuracy and identify any weaknesses.
   - **Update Membership Functions and Fuzzy Rules:** Adjust the membership functions and rules based on insights derived from real data to ensure a better representation of reality.
4. **Apply the Model as a Decision Support Tool**
   - **Integrate the Model into Enterprise Planning Systems:** Use the model for production planning, inventory management, and pricing strategies.
   - **Train Teams on Model Usage:** Ensure operational teams understand how to interpret the model's results and incorporate them into decision-making processes.
5. **Dynamically Respond to Market Changes**
   - **Regularly Update Data:** Continuously collect market data to keep the model aligned with current changes.
   - **Develop Advanced Simulation Scenarios:** Create multiple simulation scenarios that include uncertainties and significant market fluctuations.
6. **Improve Integration Between Departments**
   - **Enhance Communication Among Marketing, Production, and Finance Teams:** Ensure that strategies and decisions are consistent with the model's findings.
   - **Use the Model to Enhance Coordination:** Employ the results as a shared tool to understand and synchronize efforts across departments.
7. **Focus on Customer Satisfaction and Enhance User Experience**
   - **Leverage Results to Improve Customer Service:** Use demand forecasts to better meet customer needs.
   - **Customize Offers and Marketing Strategies:** Tailor marketing efforts based on a deeper understanding of customer behavior and anticipated demand.

8.2 General Conclusions

- **Effectiveness of the Intelligent Model:** The model has demonstrated its capability to analyze complex relationships among various factors—demand, supply, and cost—and to provide reliable results that support decision-making processes.
- **Handling Uncertainty:** By employing fuzzy logic, the model effectively manages ambiguous and uncertain data, enhancing its accuracy and flexibility.
- **Importance of Balancing Supply and Demand:** The different scenarios revealed that achieving a balance between supply and demand is crucial for superior performance, while imbalances can lead to lost opportunities or wasted resources.
- **Impact of Costs on Profitability:** Controlling costs is essential for maintaining profit margins and strengthening competitive advantage in the market.
- **Need to Adapt to Market Changes:** The simulations showed that quickly responding to shifts in demand and economic factors is vital for sustaining high performance.
- **Applicability Across Various Fields:** The model can be applied in multiple industries and sectors, broadening its utility as an analytical tool.

This study represents a significant effort in developing an intelligent model based on fuzzy logic to analyze and predict the performance of economic systems. Through simulation and in-depth analysis, the model has proven its ability to provide valuable insights that support strategic decision-making. The recommendations aim to enhance the model's effectiveness and practical application, emphasizing the importance of defining inputs based on real factors, ensuring data accuracy, and continuously adapting to market dynamics. The model can serve as a powerful tool for companies and organizations seeking to improve their performance and compete effectively in complex and ever-changing business environments.

References

Bezděk, V. (2014). Using fuzzy logic in business. Procedia-Social and Behavioral Sciences, 124, 371-380.

Carlino, G. A. (2012). Economies of scale in manufacturing location: theory and measure. (Vol. 12). Springer Science & Business Media.

Carlsson, C. F. (2012). Fuzzy logic in management . Springer Science & Business Media., Vol. 66.

Christopher, M. &. (2017). Supply chain 2.0 revisited: a framework for managing volatility-induced risk in the supply chain. . International Journal of Physical Distribution & Logistics Management, , 47(1), 2-17.

De Villiers, C. &. (2017). Integrated reporting. Sustainability accounting and integrated reporting, 13-24.

Färe, R. G. (1994). Productivity growth, technical progress, and efficiency change in industrialized countries. The American economic review, 66-83.

Forgács, A. L. (2021). The investigation of the applicability of fuzzy rule-based systems to predict economic decision-making. Acta Polytechnica Hungarica, , 18(11), 97-115.

Ganga, G. M. (2011). A fuzzy logic approach to supply chain performance management. . International Journal of Production Economics, 134(1), 177-187.

Gavetti, G. G. (2012). The behavioral theory of the firm: Assessment and prospects. Academy of Management Annals, 6(1), 1-40.

Hall, R. E. (1988). The relation between price and marginal cost in US industry. Journal of political Economy,, 96(5), 921-947.

Jeffers, J. R. (1971). On the demand versus need for medical services and the concept of" shortage". American Journal of Public Health,, 61(1), 46-63.

Kauder, E. (2015). History of marginal utility theory (Vol. (Vol. 2238)). Princeton University Press.

Korol, T. (2011). An evaluation of effectiveness of fuzzy logic model in predicting the business bankruptcy. Romanian Journal of Economic Forecasting, 92-107.

Kumar, D. S. (2013). A fuzzy logic based decision support system for evaluation of suppliers in supply chain management practices. Mathematical and Computer Modelling,, 58(11-12), 1679-1695.

Laibman, D. (2019). Is There a Classical Theory of Supply and Demand?. In Growth, Distribution and Effective Demand. New York: Routledge.

Lysons, K. &. (2020). Procurement and supply chain management. Pearson UK.

Nunes, I. L.-M. (2012). Applications of fuzzy logic in risk assessment–The RA_X case. Fuzzy inference system-Theory and applications,, 21-40.

Shapiro, A. F. (2015). Risk assessment applications of fuzzy logic. Casualty Actuarial Society,. Canadian Institute of Actuaries, Society of Actuaries, 390.

Shephard, R. W. (2015). Theory of cost and production functions. Princeton University Press.

Stojić, G. (2012). Using fuzzy logic for evaluating the level of countries'(regions') economic development. Panoeconomicus, 59(3), 293-310.

Syntetos, A. A. (2009). Forecasting for inventory planning: a 50-year review. . Journal of the Operational Research Society, , M.60(sup1), S149-S160.

Taghipour, A. (. (2020). Demand forecasting and order planning in supply chains and humanitarian logistics. IGI Global.

Voynarenko, M. P. (2021). Applying fuzzy logic to modeling economic emergence. . WSEAS Transactions on Business and Economics, Vol. 18: 424-434.

Whinston, M. D. (1995). Microeconomic theory. Oxford University Press. Caps, 3(5), 15.

Witt, U. (2001). Learning to consume–A theory of wants and the growth of demand. . Journal of evolutionary economics., 11, 23-36.

Wu, P. J. (2017). The impact of carbon emission costs on manufacturers' production and location decision. International Journal of Production Economics, , 193, 193-206.

Zadeh, G. J. (vol. 3 (1971)). Fuzzy sets. Information and control, vol. 8 (1965), pp. 338–353. - L. A. Zadeh. Similarity relations and fuzzy orderings. Information sciences, . Journal of Symbolic Logic., pp. 177–200.

Zadeh, L. A. ((1996)). Fuzzy sets, fuzzy logic, and fuzzy systems. World scientific., selected papers (Vol. 6).