The following is a decision-making program for moving floating net cages (FNC) using the Python programming language.

**PROGRAM**

import skfuzzy as fuzz

import numpy as np

# Temperature

temperatur = np.arange(0, 41, 1)

cold = fuzz.gauss2mf(temperatur, 2.782, 3.434, 17.7, 2.3)

cool = fuzz.gaussmf(temperatur, 24, 1.7)

normal = fuzz.gaussmf(temperatur, 28.5, 1.7)

warm = fuzz.gaussmf(temperatur, 33, 1.7)

hot = fuzz.gauss2mf(temperatur, 38.63, 2.278, 41, 0.034)

# Dissolved Oxygen

dissolved\_oxygen = np.arange(0, 10, 0.1)

very\_poor = fuzz.gauss2mf(dissolved\_oxygen, 0, 0.007, 2.1, 0.28)

poor = fuzz.gaussmf(dissolved\_oxygen, 2.8, 0.2)

normal\_DO = fuzz.gaussmf(dissolved\_oxygen, 3.5, 0.2)

good = fuzz.gaussmf(dissolved\_oxygen, 4.2, 0.2)

very\_good = fuzz.gauss2mf(dissolved\_oxygen, 5, 0.28, 10, 0.007)

# pH

pH = np.arange(0, 14, 0.1)

very\_Acidid = fuzz.gauss2mf(pH, 0, 0.0119, 4, 0.5535)

Acidid = fuzz.gaussmf(pH, 5.5, 0.5)

normal\_pH = fuzz.gaussmf(pH, 7.5, 0.5)

Alkaline = fuzz.gaussmf(pH, 9.5, 0.5)

very\_Alkaline = fuzz.gauss2mf(pH, 11, 0.5535, 14, 0.0119)

# Visualize the membership functions

import matplotlib.pyplot as plt

fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8, 9))

ax0.plot(temperatur, cold, 'b', linewidth=1.5, label='Cold')

ax0.plot(temperatur, cool, 'g', linewidth=1.5, label='Cool')

ax0.plot(temperatur, normal, 'r', linewidth=1.5, label='Normal')

ax0.plot(temperatur, warm, 'm', linewidth=1.5, label='Warm')

ax0.plot(temperatur, hot, 'y', linewidth=1.5, label='Hot')

ax0.set\_title('Temperature')

ax0.legend()

ax1.plot(dissolved\_oxygen, very\_poor, 'b', linewidth=1.5, label='Very Poor')

ax1.plot(dissolved\_oxygen, poor, 'g', linewidth=1.5, label='Poor')

ax1.plot(dissolved\_oxygen, normal\_DO, 'r', linewidth=1.5, label='Normal')

ax1.plot(dissolved\_oxygen, good, 'm', linewidth=1.5, label='Good')

ax1.plot(dissolved\_oxygen, very\_good, 'y', linewidth=1.5, label='Very Good')

ax1.set\_title('Dissolved Oxygen')

ax1.legend()

ax2.plot(pH, very\_Acidid, 'b', linewidth=1.5, label='Very Acidid')

ax2.plot(pH, Acidid, 'g', linewidth=1.5, label='Acidid')

ax2.plot(pH, normal\_pH, 'r', linewidth=1.5, label='Normal')

ax2.plot(pH, Alkaline, 'm', linewidth=1.5, label='Alkaline')

ax2.plot(pH, very\_Alkaline, 'y', linewidth=1.5, label='Very Alkaline')

ax2.set\_title('pH')

ax2.legend()

def decision\_making(temperature\_value, dissolved\_oxygen\_value, pH\_value):

if temperature\_value >= 24 and temperature\_value <= 33 and dissolved\_oxygen\_value >= 3.0 and pH\_value >= 5.5 and pH\_value <= 9.5:

return "Stay\_In\_Position"

else:

return "Move"

# Get user inputs

temperature\_value = float(input("Enter the temperature: "))

dissolved\_oxygen\_value = float(input("Enter the dissolved oxygen value: "))

pH\_value = float(input("Enter the pH: "))

# Result

result = decision\_making(temperature\_value, dissolved\_oxygen\_value, pH\_value)

print("The Floating Net Cage should be: ", result)

**ALGORITHM:**

1. Start

2. Declare the temperature variable using the range function (0, 41, 1).

3. Declare gauss2mf functions for cold, cool, normal, warm, and hot with temperature

parameters.

4. Declare the dissolved\_oxygen variable using the range function (0, 10, 0.1).

5. Declare gauss2mf functions for very\_poor, poor, normal\_DO, good, and very\_good with the parameter dissolved\_oxygen.

6. Declare the pH variable using the range function (0, 14, 0.1).

7. Declare gauss2mf functions for very\_Acidid, Acidid, normal\_pH, Alkaline, and very\_Alkaline with pH parameters.

8. Create a decision\_making function with parameters temperature\_value, dissolved\_oxygen\_value, and pH\_value.

9. If the temperature value is between 24°C and 33°C, the dissolved oxygen value is ≥3.0 and the pH value is between 5.5 and 9.5, then the floating net cages (FNC) status is “Stay in Position”.

10. If not, then the FNC status is "Move".

11. Ask for user input for temperature\_value, dissolved\_oxygen\_value, and pH\_value

12. Enter the decision\_making function with parameters temperature\_value, dissolved\_oxygen\_value, and pH\_value.

13. Calculate the result with the decision\_making function.

14. Display the result.

15. done

The following is the opensource software used and the access link.

**Software used:**

1. Anaconda Navigator ver 2.4.0

<https://docs.anaconda.com/anaconda/install/hashes/win-3-64/>

2. Jupyter Notebook ver 6.5.2

3. python ver 3.9.16

4. numpy ver 1.24.2

<https://pypi.org/project/numpy/>

5. sci-kit fuzzy ver 0.4.2

<https://pypi.org/project/scikit-fuzzy/>

6. matplotlib ver 3.7.1

<https://pypi.org/project/matplotlib/>