1. The SIS-DRI model

import numpy as np

matrix1 = np.array([[0.5, 0.3, 0.8, 0.5],

[0.7, 0.5, 0.6, 0.4],

[0.2, 0.4, 0.5, 0.3],

[0.5, 0.6, 0.7, 0.5]])

matrix2 = np.array([[0.5, 0.7, 0.6, 0.6],

[0.3, 0.5, 0.5, 0.4],

[0.4, 0.5, 0.5, 0.7],

[0.4, 0.6, 0.3, 0.5]])

matrix3 = np.array([[0.5, 0.4, 0.3, 0.5],

[0.6, 0.5, 0.4, 0.3],

[0.7, 0.6, 0.5, 0.2],

[0.5, 0.7, 0.8, 0.5]])

matrix4 = np.array([[0.5, 0.6, 0.9, 0.3],

[0.4, 0.5, 0.7, 0.4],

[0.1, 0.3, 0.5, 0.9],

[0.7, 0.6, 0.1, 0.5]])

eta\_values = [1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8]

weights = ([[0.08, 0.03, 0.84, 0.05],

[0.13, 0.07, 0.71, 0.09],

[0.18, 0.1, 0.59, 0.13],

[0.21, 0.13, 0.5, 0.16],

[0.23, 0.16, 0.42, 0.19],

[0.24, 0.19, 0.35, 0.22],

[0.25, 0.22, 0.3, 0.23]])

modified\_matrices1 = []

best\_eta = -1

best\_delta = -1

best\_consistency\_degree = -1

best\_TCL = -1

best\_final\_value1 = float('-inf')

best\_final\_value2 = float('-inf')

for eta in eta\_values:

group\_decision\_matrix = matrix1 \* weights[int(eta\*8-1)][1] + matrix2 \* weights[int(eta\*8-1)][0] + matrix3 \* weights[int(eta\*8-1)][3] +matrix4 \* weights[int(eta\*8-1)][2]

print(f"For η={eta}:")

print("Collective preference relation:")

print(group\_decision\_matrix)

delta\_values = np.linspace(0, 1, 101)

delta\_values = delta\_values[1:-1]

for delta in delta\_values:

modified\_matrix1 = delta \* matrix3 + (1 - delta) \* group\_decision\_matrix

modified\_matrices1.append(modified\_matrix1)

consensus\_degree1 = np.sum(abs(modified\_matrix1 - (1/4) \* (matrix1 + matrix2 + modified\_matrix1 + modified\_matrix2)))

n = 4

consensus1 = 1 - (consensus\_degree1 / (n \* (n - 1)))

consensus = consensus1

print(f"For δ={delta}:")

print("Preference relation of g3 after the CRP:")

print(modified\_matrix1)

print("Consensus level of g3 after CRP", consensus1)

modification\_cost1 = np.abs(matrix3 - modified\_matrix1)

diff\_sum1 = np.sum(modification\_cost1)

cost1 = 1 - (diff\_sum1 / (n \* (n - 1)))

cost =cost1

print("Consensus cost of g3:", cost1)

consistency\_degree1 = abs(modified\_matrix1[0, 1] + modified\_matrix1[1, 2] - modified\_matrix1[0, 2] - 0.5) \ + abs(modified\_matrix1[1, 2] + modified\_matrix1[2, 3] - modified\_matrix1[1, 3] - 0.5) \+ abs(modified\_matrix1[2, 3] + modified\_matrix1[3, 0] - modified\_matrix1[2, 0] - 0.5) \+ abs(modified\_matrix1[3, 0] + modified\_matrix1[0, 1] - modified\_matrix1[3, 1] - 0.5)

consistency\_degree1 = 1 - 4 / (n \* (n - 1) \* (n - 2)) \* abs(consistency\_degree1)

consistency\_degree = consistency\_degree1

print("Consistency level of g3 after CRP:", consistency\_degree1)

print()

alpha = 0.5

TCL = alpha \* consensus + (1 - alpha) \* (1 - cost )

print("TCL :",TCL)

if consistency\_degree > best\_consistency\_degree and TCL > 0.5:

best\_consistency\_degree = consistency\_degree

best\_eta = eta

best\_delta = delta

best\_TCL = TCL

best\_final\_value1 = modified\_matrix1[3, 0] + modified\_matrix1[0, 1]

best\_final\_value2 = modified\_matrix2[3, 0] + modified\_matrix2[0, 1]

print("-" \* 20)

print(f"The maximum consistency : {best\_consistency\_degree}")

print(f"η: {best\_eta}")

print(f"δ: {best\_delta}")

2.The SEN-DRI model

import numpy as np

matrix1 = np.array([[0.5, 0.3, 0.8, 0.5],

[0.7, 0.5, 0.6, 0.4],

[0.2, 0.4, 0.5, 0.3],

[0.5, 0.6, 0.7, 0.5]])

matrix2 = np.array([[0.5, 0.7, 0.6, 0.6],

[0.3, 0.5, 0.5, 0.4],

[0.4, 0.5, 0.5, 0.7],

[0.4, 0.6, 0.3, 0.5]])

matrix3 = np.array([[0.5, 0.4, 0.3, 0.5],

[0.6, 0.5, 0.4, 0.3],

[0.7, 0.6, 0.5, 0.2],

[0.5, 0.7, 0.8, 0.5]])

matrix4 = np.array([[0.5, 0.6, 0.9, 0.3],

[0.4, 0.5, 0.7, 0.4],

[0.1, 0.3, 0.5, 0.9],

[0.7, 0.6, 0.1, 0.5]])

eta\_values = [1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8]

weights = ([[0.08, 0.03, 0.84, 0.05],

[0.13, 0.07, 0.71, 0.09],

[0.18, 0.1, 0.59, 0.13],

[0.21, 0.13, 0.5, 0.16],

[0.23, 0.16, 0.42, 0.19],

[0.24, 0.19, 0.35, 0.22],

[0.25, 0.22, 0.3, 0.23]])

modified\_matrices1 = []

best\_eta = -1

best\_delta = -1

best\_consistency\_degree = -1

best\_TCL = -1

best\_final\_value1 = float('-inf')

best\_final\_value2 = float('-inf')

for eta in eta\_values:

group\_decision\_matrix = matrix1 \* weights[int(eta\*8-1)][1] + matrix2 \* weights[int(eta\*8-1)][0] + matrix3 \* weights[int(eta\*8-1)][3] +matrix4 \* weights[int(eta\*8-1)][2]

print(f"For η={eta}:")

print("Collective preference relation:")

print(group\_decision\_matrix)

delta\_values = np.linspace(0, 1, 101)

delta\_values = delta\_values[1:-1]

for delta in delta\_values:

modified\_matrix1 = delta \* matrix3 + (1 - delta) \* group\_decision\_matrix

modified\_matrices1.append(modified\_matrix1)

consensus\_degree1 = np.sum(abs(modified\_matrix1 - (1/4) \* (matrix1 + matrix2 + modified\_matrix1 + modified\_matrix2)))

n = 4

consensus1 = 1 - (consensus\_degree1 / (n \* (n - 1)))

consensus = consensus1

print(f"For δ={delta}:")

print("Preference relation of g3 after the CRP:")

print(modified\_matrix1)

print("Consensus level of g3 after CRP", consensus1)

modification\_cost1 = np.abs(matrix3 - modified\_matrix1)

diff\_sum1 = np.sum(modification\_cost1)

cost1 = 1 - (diff\_sum1 / (n \* (n - 1)))

cost =cost1

print("Consensus cost of g3:", cost1)

consistency\_degree1 = abs(modified\_matrix1[0, 1] + modified\_matrix1[1, 2] - modified\_matrix1[0, 2] - 0.5) + abs(modified\_matrix1[1, 2] + modified\_matrix1[2, 3] - modified\_matrix1[1, 3] - 0.5) + abs(modified\_matrix1[2, 3] + modified\_matrix1[3, 0] - modified\_matrix1[2, 0] - 0.5) + abs(modified\_matrix1[3, 0] + modified\_matrix1[0, 1] - modified\_matrix1[3, 1] - 0.5)

consistency\_degree1 = 1 - 4 / (n \* (n - 1) \* (n - 2)) \* abs(consistency\_degree1)

consistency\_degree = consistency\_degree1

print("Consistency level of g3 after CRP:", consistency\_degree1)

print()

beta = 0.5

TCI = beta \* consistency + (1 - beta) \* (1 - cost )

print("TCI :",TCI)

if consensus\_degree > best\_consensus\_degree and TCI > 0.5:

best\_consensus\_degree =consensus\_degree

best\_eta = eta

best\_delta = delta

best\_TCI = TCI

best\_final\_value1 = modified\_matrix1[3, 0] + modified\_matrix1[0, 1]

best\_final\_value2 = modified\_matrix2[3, 0] + modified\_matrix2[0, 1]

print("-" \* 20)

print(f"The maximum consensus : {best\_consensus\_degree}")

print(f"η: {best\_eta}")

print(f"δ: {best\_delta}")

3.The CO-DRI model

import numpy as np

matrix1 = np.array([[0.5, 0.3, 0.8, 0.5],

[0.7, 0.5, 0.6, 0.4],

[0.2, 0.4, 0.5, 0.3],

[0.5, 0.6, 0.7, 0.5]])

matrix2 = np.array([[0.5, 0.7, 0.6, 0.6],

[0.3, 0.5, 0.5, 0.4],

[0.4, 0.5, 0.5, 0.7],

[0.4, 0.6, 0.3, 0.5]])

matrix3 = np.array([[0.5, 0.4, 0.3, 0.5],

[0.6, 0.5, 0.4, 0.3],

[0.7, 0.6, 0.5, 0.2],

[0.5, 0.7, 0.8, 0.5]])

matrix4 = np.array([[0.5, 0.6, 0.9, 0.3],

[0.4, 0.5, 0.7, 0.4],

[0.1, 0.3, 0.5, 0.9],

[0.7, 0.6, 0.1, 0.5]])

eta\_values = [1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8]

weights = ([[0.08, 0.03, 0.84, 0.05],

[0.13, 0.07, 0.71, 0.09],

[0.18, 0.1, 0.59, 0.13],

[0.21, 0.13, 0.5, 0.16],

[0.23, 0.16, 0.42, 0.19],

[0.24, 0.19, 0.35, 0.22],

[0.25, 0.22, 0.3, 0.23]])

modified\_matrices1 = []

best\_eta = -1

best\_delta = -1

best\_consistency\_degree = -1

best\_TCL = -1

best\_final\_value1 = float('-inf')

best\_final\_value2 = float('-inf')

for eta in eta\_values:

group\_decision\_matrix = matrix1 \* weights[int(eta\*8-1)][1] + matrix2 \* weights[int(eta\*8-1)][0] + matrix3 \* weights[int(eta\*8-1)][3] +matrix4 \* weights[int(eta\*8-1)][2]

print(f"For η={eta}:")

print("Collective preference relation:")

print(group\_decision\_matrix)

delta\_values = np.linspace(0, 1, 101)

delta\_values = delta\_values[1:-1]

for delta in delta\_values:

modified\_matrix1 = delta \* matrix3 + (1 - delta) \* group\_decision\_matrix

modified\_matrices1.append(modified\_matrix1)

consensus\_degree1 = np.sum(abs(modified\_matrix1 - (1/4) \* (matrix1 + matrix2 + modified\_matrix1 + modified\_matrix2)))

n = 4

consensus1 = 1 - (consensus\_degree1 / (n \* (n - 1)))

consensus = consensus1

print(f"For δ={delta}:")

print("Preference relation of g3 after the CRP:")

print(modified\_matrix1)

print("Consensus level of g3 after CRP", consensus1)

modification\_cost1 = np.abs(matrix3 - modified\_matrix1)

diff\_sum1 = np.sum(modification\_cost1)

cost1 = 1 - (diff\_sum1 / (n \* (n - 1)))

cost =cost1

print("Consensus cost of g3:", cost1)

consistency\_degree1 = abs(modified\_matrix1[0, 1] + modified\_matrix1[1, 2] - modified\_matrix1[0, 2] - 0.5) \

+ abs(modified\_matrix1[1, 2] + modified\_matrix1[2, 3] - modified\_matrix1[1, 3] - 0.5) \

+ abs(modified\_matrix1[2, 3] + modified\_matrix1[3, 0] - modified\_matrix1[2, 0] - 0.5) \

+ abs(modified\_matrix1[3, 0] + modified\_matrix1[0, 1] - modified\_matrix1[3, 1] - 0.5)

consistency\_degree1 = 1 - 4 / (n \* (n - 1) \* (n - 2)) \* abs(consistency\_degree1)

consistency\_degree = consistency\_degree1

print("Consistency level of g3 after CRP:", consistency\_degree1)

print()

nu = 0.5

CLI =nu \* consistency + (1 - nu) \* (1 - cost )

print(" CLI :", CLI )

if cost\_degree < best\_cost\_degree and CLI > 0.8:

best\_cost\_degree =cost\_degree

best\_eta = eta

best\_delta = delta

best\_ CLI = CLI

best\_final\_value1 = modified\_matrix1[3, 0] + modified\_matrix1[0, 1]

best\_final\_value2 = modified\_matrix2[3, 0] + modified\_matrix2[0, 1]

print("-" \* 20)

print(f"The minimum cost : {best\_cost\_degree}")

print(f"η: {best\_eta}")

print(f"δ: {best\_delta}")