Batch: A1 **Roll No.:** 16010120015

Experiment No. 08

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Defuzzification methods.

Aim : To understand the concept of Defuzzification.

Expected Outcome of Experiment:

CO4 : Apply basics of Fuzzy logic and neural networks

Books/ Journals/ Websites referred:

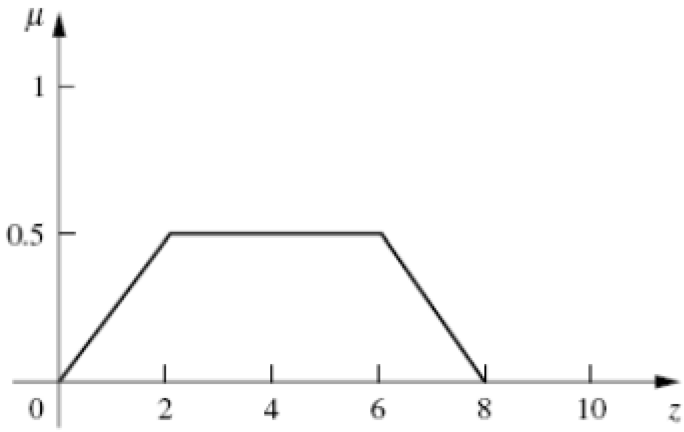
- J.S.R.Jang, C.T.Sun and E.Mizutani, "Neuro-Fuzzy and Soft Computing", PHI, 2004, Pearson Education 2004.
- Davis E.Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning" Addison Wesley, N.Y., 1989.
- S. Rajasekaran and G.A.V.Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI, 2003.
- http://library.thinkquest.org/C007395/tqweb/history.html

Pre Lab/ Prior Concepts:

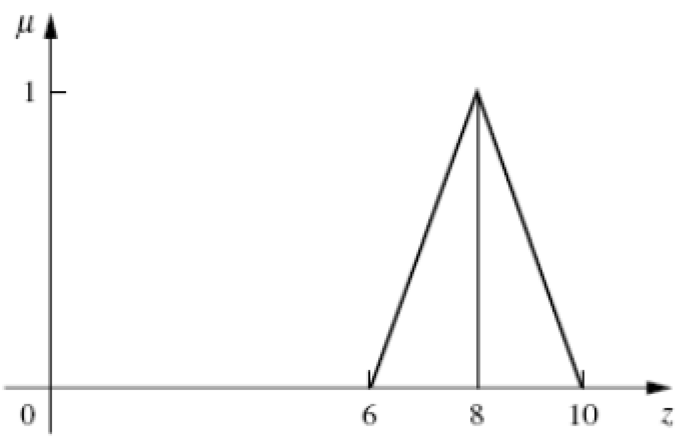
Defuzzification :

Defuzzification is the process of producing a quantifiable result in Crisp logic, given fuzzy sets and corresponding membership degrees. It is the process that maps a fuzzy set to a crisp set. It is typically needed in fuzzy control systems. These will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets. Defuzzification is the conversion of a fuzzy quantity to a precise quantity, just as fuzzification is the conversion of a precise quantity to a fuzzy quantity. μ

For example, **Fig (a)** shows the first part of the Fuzzy output and **Fig (b)** shows the second part of the Fuzzy output.

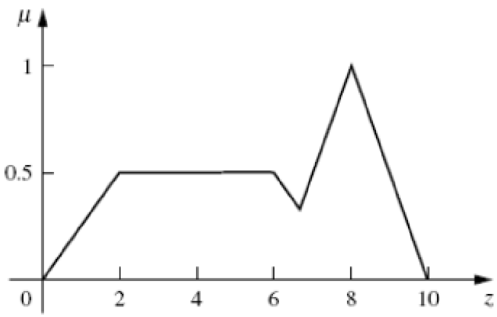


(*a*)



(*b*)

Then **Fig (c)** shows the union of the two parts (a) and (b).



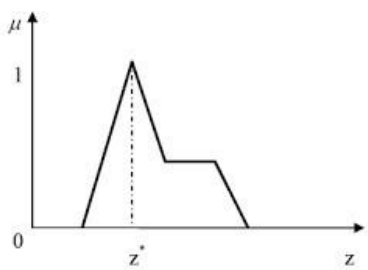
(c)

Different Defuzzification methods

1. **Max membership method**

This method is also known as height method and is limited to peak output functions. This method is given by the algebraic expression:

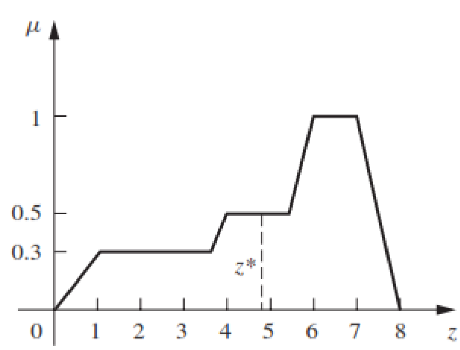
μ(z*) >= **μ**(z) for all z ∊ Z.



2. **Center of gravity or centroid**

This method is also known as the centre of mass, centre of area or centre of gravity. It is the most commonly used defuzzification method. The defuzzified output z* is given by:

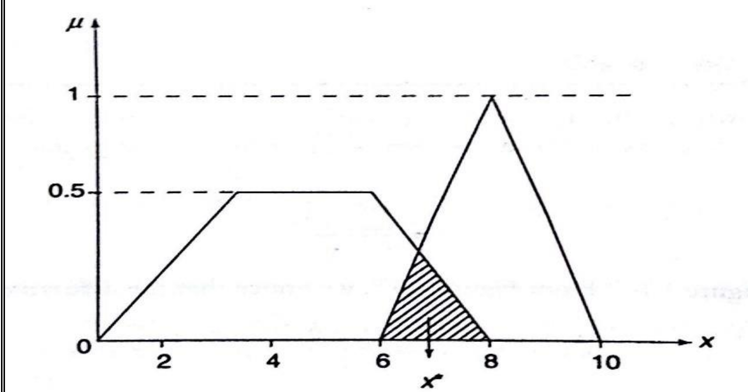
z* = ∫μ(z).zdz / ∫μ(z)dz



3. Centre of sums

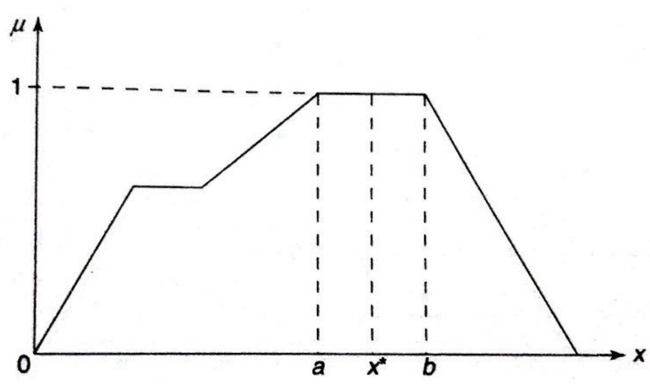
This method employs the algebraic sum of the individual fuzzy subsets instead of their union. The calculations here are very fast, but the main drawback is that the intersecting areas are added twice. The defuzzified value z* is given by

$z^* = \int z^* \sum \mu(z).z dz / \int \sum \mu(z) dz$



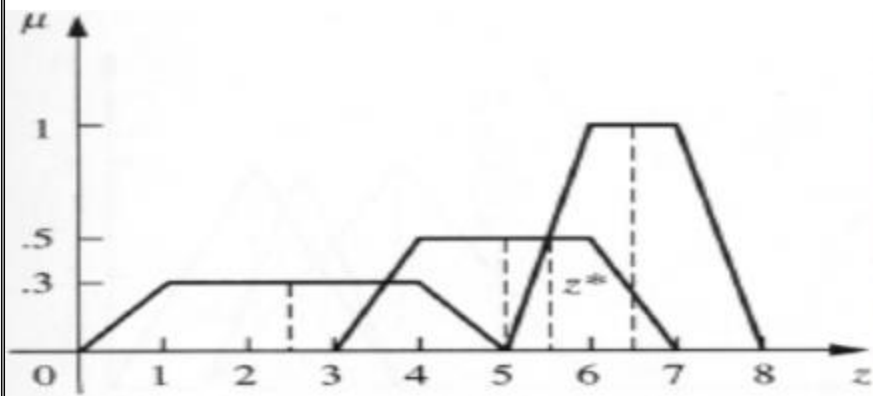
4. Mean of maximum method

This method is also known as the middle of the maxima. This is closely related to the max-membership method, except that the locations of the maximum membership can be nonunique. The output here is given by:
$z^* = \sum z' / n$; where z' is the maximum value of the membership function.



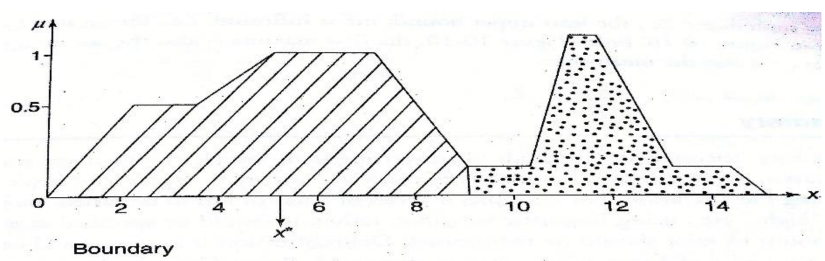
5. Weighted average method

This method is valid for symmetrical output membership functions only. Each membership function is weighted by its maximum membership value. The output in the case is given by $z^* = \sum\mu(z').z' / \sum\mu(z')$; where z' is the maximum value of the membership function.



6. **Centre of Largest Area**
   This method can be adopted when the output of at least two convex fuzzy subsets which are not overlapping. The output, in this case, is biased towards a side of one membership function. When output fuzzy st has at least two convex regions, then the centre of gravity of the convex fuzzy subregion having the largest are is used to obtain the defuzzified value z*. The value is given by

   $z^* = \int \mu c(z).zdz / \int \sum\mu c(z)dz$



Implementation Details:

```
from matplotlib import pyplot as plt
import numpy as np
from numpy import trapz

#Trapezoidal Membership Function
def trapez(x,a,b,c,d):
  if(x<a):
    return 0
  elif(x>=a and x<=b):
    return float((x)-(a))/float((b)-(a))
  elif(x>=b and x<=c):
    return 1
  elif(x>=c and x<=d):
    return float((d)-(x))/float((d)-(c))
  elif(x>d):
    return 0

#Triangular Membership Function
def tri(x,a,b,c):
  if(x<=a):
    return 0
  elif(x>=a and x<=b):
    return float((x)-(a))/float((b)-(a))
  elif(x>=b and x<=c):
    return float((c)-(x))/float((c)-(b))
  else:
    return 0

#Union Membership Function
def Union(X):
  trapezium_array=[]
  triangle_array=[]
  for i in range(500):
    trapezium_array.append(trapez(X[i],4,7,8,9))
    triangle_array.append(tri(X[i],1,5,9))
  union_array=[]
  for i in range(500):
    union_array.append(max(triangle_array[i],trapezium_array[i]))
  return union_array
```

```
#Centre Of Gravity Membership Function
def Centre_of_Grav():
  C=Union(X)
  centroid=0.0
  sum=0.0
  for i in range(500):
    sum+=C[i]
  centroid=float(sum/500)
  xmax = centre1(C)
  ymax = np.interp(xmax, X,C)
  fig = plt.figure()
  ax = fig.add_subplot(111)
  plt.plot(X,C,label = "C")
  ax.annotate('Centre of Gravity (Z*)', xy=(xmax, ymax),
xytext=(xmax,ymax+0.),arrowprops=dict(facecolor='green'))
  ax.set_ylim(0,2)
  plt.show()
  print("Centre of Gravity: "+str(xmax))

  #Centre Of sum Membership Function
def Centre_of_Sum():
  trapezium_1=[]
  triangle_1=[]
  for i in range(500):
    trapezium_1.append(trapez(X[i],4,5,7,9))
    triangle_1.append(tri(X[i],1,3,9))
  sum1=0
  sum2=0
  area1=trapz(trapezium_1, dx=7)
  area2=trapz(triangle_1, dx=6)
  z1=4.5
  z2= 7
  sum1=(area1*z1)+(area2*z2)
  sum2=area1+area2
  centre=float(sum1/sum2)
  xmax = centre
  ymax = np.interp(centre, X,triangle_1)
  fig = plt.figure()
  ax = fig.add_subplot(111)
  plt.plot(X,trapezium_1,label = "Trapezium 1")
  plt.plot(X,triangle_1, label = "Trapezium 2")
  plt.legend()
```

```
  ax.annotate('Centre of Sums (Z*)', xy=(xmax, ymax),
xytext=(xmax,ymax+0.5),arrowprops=dict(facecolor='green'))
  ax.set_ylim(0,2)
  plt.show()
  print("Centre of Sums: "+str(centre))

  #Mean of Maximum Membership Function
def Mean_of_Max():
  trapezium_1=[]
  trapezium_2=[]
  X=np.linspace(1,40,500)
  for i in range(500):
    trapezium_1.append(trapez(X[i],2,15,22,29))
    trapezium_2.append(trapez(X[i],22,37,45,39))
  C=[]
  for i in range(500):
    C.append(max(trapezium_1[i],trapezium_2[i]))
  sum=0
  x_values=[]
  for i in range(500):
    if(C[i]==1):
      x_values.append(X[i])
  a=x_values[0]
  b=x_values[-1]
  mean=float(a+b)/2
  fig = plt.figure()
  ax = fig.add_subplot(111)
  plt.plot(X,C,label = "C")
  xmax = mean
  ymax = np.interp(mean, X,C)
  a_x=a
  a_y= np.interp(a, X,C)
  b_x=b
  b_y= np.interp(b, X,C)
  ax.annotate('Mean of Maximum (Z*)', xy=(xmax, ymax),
xytext=(xmax,ymax+0.5),arrowprops=dict(facecolor='green'))

  ax.annotate('a', xy=(a_x, a_y),
xytext=(a_x,a_y+0.5),arrowprops=dict(facecolor='green'))

  ax.annotate('b', xy=(b_x, b_y),
xytext=(b_x,b_y+0.35),arrowprops=dict(facecolor='green'))
```

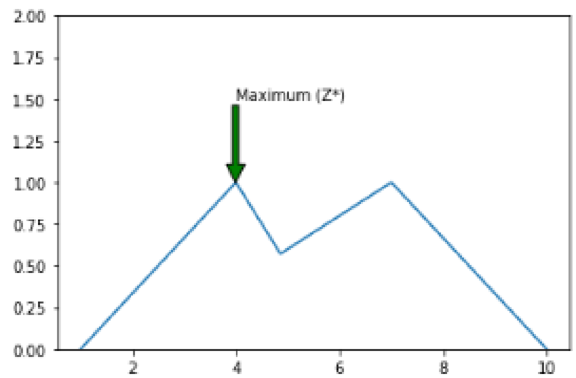
```
  ax.set_ylim(0,1.7)
  plt.show()
  print("Mean of Maximum: "+str(mean))

#Average Weight Membership Function
def Weighted_Avg():
  trapezium_1=[]
  trapezium_2=[]
  for i in range(500):
    trapezium_1.append(trapez(X[i],1,4,2,9))
    trapezium_2.append(trapez(X[i],3,4,6,10))
  C=[]
  for i in range(500):
    C.append(max(trapezium_1[i],trapezium_2[i]))
  sum=0
  a= 4.5
  b= 6.5
  mean=float(a+b)/2
  fig = plt.figure()
  ax = fig.add_subplot(111)
  plt.plot(X,trapezium_1,label = "Trapezium 1")
  plt.plot(X,trapezium_2, label = "Trapezium 2")
  xmax = mean
  ymax = np.interp(mean, X,C)
  a_x=a
  a_y= np.interp(a, X,C)
  b_x=b
  b_y= np.interp(b, X,C)
  ax.annotate('Weighted Average (Z*)', xy=(xmax, ymax), xytext=(xmax,

  ymax+0.58),arrowprops=dict(facecolor='green'))

  ax.annotate('a', xy=(a_x, a_y), xytext=(a_x,

  a_y+0.5),arrowprops=dict(facecolor='green'))

  ax.annotate('b', xy=(b_x, b_y),
xytext=(b_x,b_y+0.35),arrowprops=dict(facecolor='green'))
  ax.set_ylim(0,1.7)
  plt.show()
  print("Weighted Average: "+str(mean))
```

```
  def centre1(C):
  sum1=0
  sum2=0
  for i in range(500):
    sum1+=C[i]*X[i]
    sum2+=C[i]
  centre=float(sum1/sum2)
  return centre


def centre2(C):
  sum1=0
  sum2=0
  for i in range(500):
    sum1+=C*X[i]
    sum2+=C
  centre=float(sum1/sum2)
  return centre


#Centre of Large Area
def Centre_of_Larg_Area():
  trapezium_1=[]
  trapezium_2=[]
  centre=0
  for i in range(500):
    trapezium_1.append(trapez(X[i],1,2,3,4))
    trapezium_2.append(trapez(X[i],4,7,8,9))
  C=[]
  for i in range(500):
    C.append(max(trapezium_1[i],trapezium_2[i]))
  sum=0
  trapezium_1_area=trapz(trapezium_1, dx=4)
  trapezium_2_area=trapz(trapezium_2, dx=4)
  if(trapezium_1_area>trapezium_2_area):
    centre=centre2(trapezium_1_area)
    plot=trapezium_1
  else:
    centre=centre2(trapezium_2_area)
    plot=trapezium_2
  fig = plt.figure()
  ax = fig.add_subplot(111)
```

```
  plt.plot(X,trapezium_1,label = "Trapezium 1")
  plt.plot(X,trapezium_2, label = "Trapezium 2")
  xmax = centre
  ymax = np.interp(centre, X,plot)
  ax.annotate('Centre of Largest Area (Z*)', xy=(xmax, ymax),
xytext=(xmax,ymax+0.5),arrowprops=dict(facecolor='green'))
  ax.set_ylim(0,1.7)
  plt.show()
  print("Centre of Largest Area: "+str(centre))


X=np.linspace(1,10,500)
while True:
  print("Enter the following: ")
  print("1: Max-Membership")
  print("2: Centre of Gravity")
  print("3: Centre of Sums")
  print("4: Mean-Maximum")
  print("5: Weighted Average")
  print("6: Centre of Largest Area")
  print("0: Exit")
  choice=int(input("Enter your choice: "))
  if(choice==1):
    Max_Mem()
  elif(choice==2):
    Centre_of_Grav()
  elif(choice==3):
    Centre_of_Sum()
  elif(choice==4):
    Mean_of_Max()
  elif(choice==5):
    Weighted_Avg()
  elif(choice==6):
    Centre_of_Larg_Area()
  elif(choice==0):
    break
  else:
    print("Invalid Choice")
```

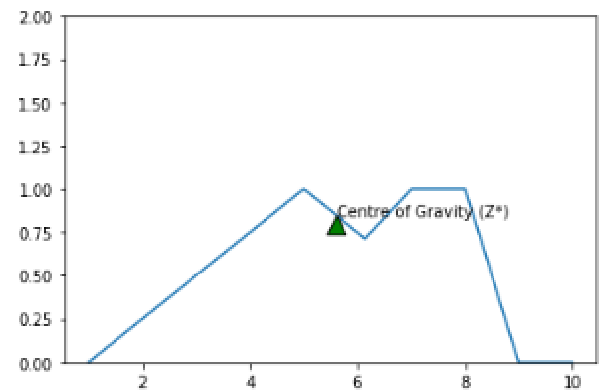
OUTPUT:

```
Enter the following:
1: Max-Membership
2: Centre of Gravity
3: Centre of Sums
4: Mean-Maximum
5: Weighted Average
6: Centre of Largest Area
0: Exit
Enter your choice: 1
```



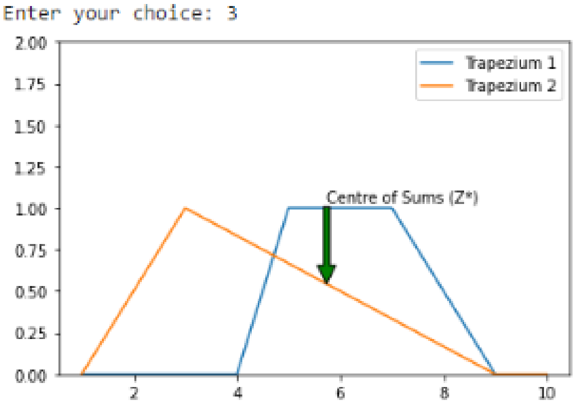

```
Maximum: 3.993987975951904
```

```
Enter the following:
1: Max-Membership
2: Centre of Gravity
3: Centre of Sums
4: Mean-Maximum
5: Weighted Average
6: Centre of Largest Area
0: Exit
Enter your choice: 2
```



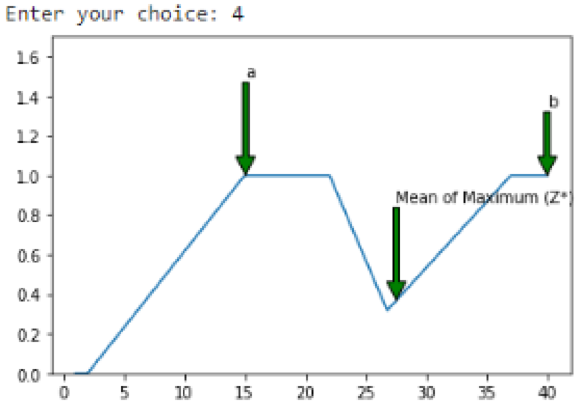

```
Centre of Gravity: 5.613847456424395
```

```
Enter the following:
1: Max-Membership
2: Centre of Gravity
3: Centre of Sums
4: Mean-Maximum
5: Weighted Average
6: Centre of Largest Area
0: Exit
Enter your choice: 3
```



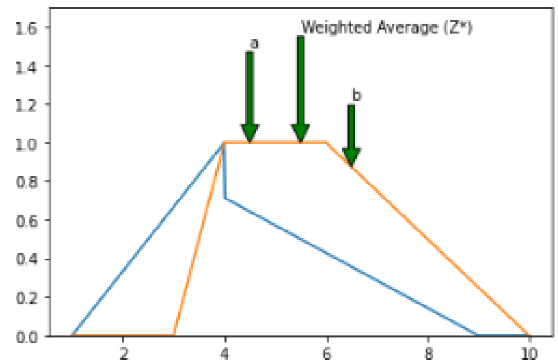

```
Centre of Sums: 5.737110981935253
```

```
Enter the following:
1: Max-Membership
2: Centre of Gravity
3: Centre of Sums
4: Mean-Maximum
5: Weighted Average
6: Centre of Largest Area
0: Exit
Enter your choice: 4
```



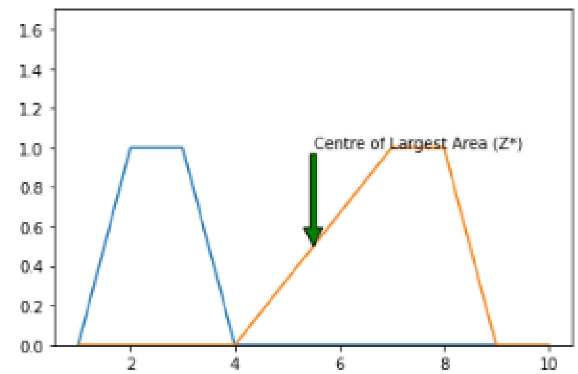

```
Mean of Maximum: 27.534068136272545
```

```
1: Max-Membership
2: Centre of Gravity
3: Centre of Sums
4: Mean-Maximum
5: Weighted Average
6: Centre of Largest Area
0: Exit
Enter your choice: 5
```




```
Weighted Average: 5.5
```

```
Enter your choice: 6
```




```
Centre of Largest Area: 5.500000000000046
```

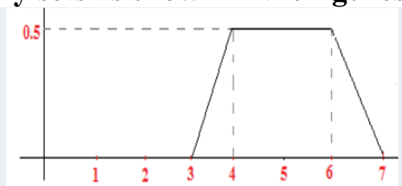
Conclusion: Thus, we have successfully implemented defuzzification methods

Lab Descriptive Questions :

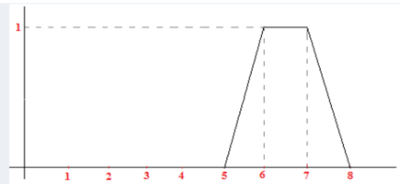
1. **Let there be 3 different fuzzy sets as shown in the figures below:-**

A

B

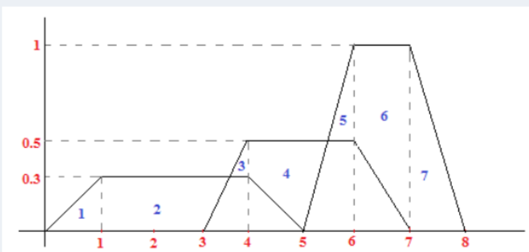


C



Hence the union of all the three sets can be represented by the following figure:-

D = A ∪ B ∪ C



Now we shall calculate (manually) the defuzzified value using all the above methods one by one.

1> Max Membership function   AI-16010120015 - YASH

→ from D we can say that the maximum membership value is 1. Hence the Scalar Value corresponding to the maximum membership value. ∴ $x^* = 6, 7$.

2> Centroid Method

In this method, we have to find the area bounded by the Union of the 3 sets

Sub Area No	Area	$\bar{x}$	Area $\times$ $\bar{x}$
1	$\frac{1 \times 0.3}{2} = 0.150$	0.67	0.100
2	$3 \times 0.3 = 0.90$	2.50	2.250
3	$\frac{0.4 \times 0.2}{2} = 0.04$	3.00	0.149
4	$2 \times 0.5 = 1.00$	~~5.87~~ 1.00	5.000
5	$\frac{0.5 \times 0.5}{2} = 0.125$	5.87	7.330
6	$1 \times 1 = 1.00$	6.50	6.500
7	$\frac{1 \times 1}{2} = 0.50$	7.33	3.600
	Σ Area = 3.715		Σ Area $\times \bar{x}$ = 24.989

$X^* = \Sigma \text{Area} \times \bar{x} \div \Sigma \text{Area}$

$X^* = 24.989 \div 3.715$

$X^* = 6.72$

3. Weighted Average Method

The Each fuzzy Set the Centre Is Calculated Individual by Multiplying the mean with its membership value & then the average of all Sets is.

Membership Value of A ~~is~~ at 2.5 = 0.3
Membership Value of B at 0.5 = 0.5
Membership Value of C at 6.5 = 1

$x^* = (2.5 \times 0.3 + 0.5 \times 1 + 6.5 \times 1) \div (0.3 + 0.5 + 1)$
$x^* = 9.75 \div 1.8$
$x^* = 5.146$

4 Centre of Sums

We Calculate the Sum of the ~~Average~~ Areas of Individual fuzzy Sets.

Area (A) = $[(5+3) \times 0.3] \div 2 = 1.2$
Area (B) = $[(4+2) \times 0.5] \div 2 = 1.5$
Area (C) = $[(3+1) \times 1] \div 2 = 2$

$x^* = [1.25 \times 2.5] + [1.5 \times 5) + (2 \times 6.5)] \div (1.2 + 1.5 + 2)$
$x^* = 5$

5 Centre of largest Area

$x^* = (6+7) \div 2$
$x^* = 6.5$

Date: 2/12/2022 **Signature of faculty in-charge**